

Universidade do Minho
Escola de Engenharia

Ângela Maria Gomes Loureiro

Transporte de cargas por equipas de robôs

Ângela Maria Gomes Loureiro Transporte de cargas por equipas de robôs



Universidade do Minho
Escola de Engenharia

Ângela Maria Gomes Loureiro

Transporte de cargas por equipas de robôs

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia Eletrónica Industrial e Computadores

Trabalho efectuado sob a orientação do
Professor Doutor Sérgio Monteiro

DECLARAÇÃO

Nome: Ângela Maria Gomes Loureiro

Endereço eletrónico: mgomesloureiro@sapo.pt Telefone: 917559047

Número do Cartão de Cidadão: 14037640

Título da dissertação:

Transporte de cargas por equipas de robôs

Orientador: Professor Doutor Sérgio Monteiro

Ano de conclusão: 2015

Dissertação submetida na Universidade do Minho para a obtenção do grau de Mestre em Engenharia Eletrónica Industrial e Computadores

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____/____/____

Assinatura: _____

Agradecimentos

Em primeiro lugar queria agradecer ao meu orientador professor Doutor Sérgio Monteiro, pelo incentivo, força, encorajamento para que desenvolvesse este trabalho de investigação. E ainda pela sua supervisão, apoio, conselhos e pela disponibilidade demonstrada ao longo da realização desta dissertação.

Um obrigado à professora Doutora Estela Bicho pela prestabilidade demonstrada em ajudar sempre que necessário.

Aos meus pais pela educação que me deram, por todo o apoio prestado durante a minha formação e por sempre acreditarem em mim.

Aos alunos de doutoramento Toni Machado e Tiago Malheiro pelo apoio, opiniões que contribuíram para a realização desta dissertação.

Um agradecimento também a todos os colegas de laboratório que criaram um ambiente agradável de trabalho e convívio.

Aos meus amigos Ricardo Gomes e Paulo Vicente pela amizade e por me apoiarem e encorajarem sempre que precisava e, especialmente, por todos os momentos que vivemos ao longo do curso.

Esta página, intencionalmente, apenas contém esta frase.

Resumo

Esta dissertação foca-se no problema de transporte de cargas por equipas de robôs móveis autónomos que atuam em coordenação. A equipa utiliza uma estratégia *leader* - *helper* e arquiteturas de controlo formalizadas recorrendo à abordagem de atratores dinâmicos para geração de comportamentos.

Em trabalhos desenvolvidos anteriormente há comunicação explícita de informação entre o robô *leader* e o robô *helper* (na forma de direções). No presente trabalho é apresentada uma solução que permite dotar o robô *helper* com capacidades de estimar essa informação e eliminar a comunicação entre ele e o robô *leader*. Esta nova contribuição irá permitir que a equipa de robôs possa movimentar-se em ambientes em que a comunicação não é permitida, adiciona robustez em ambientes em que a comunicação é ruidosa ou, ainda, permite que o robô *leader* possa ser substituído por um humano, de forma transparente do ponto de vista do robô *helper*, uma vez que terá capacidades para agir de forma independente.

Para o desenvolvimento do módulo estimador foi usado o Filtro de Kalman Estendido que utiliza informação proveniente das medições obtidas para a posição, orientação e velocidade do robô *leader*, calculadas a partir de informação sensorial do robô *helper* e do respetivo suporte de apoio à carga a transportar, para gerar uma estimativa da direção de navegação do robô *leader*.

Na validação da solução implementada realizaram-se testes experimentais tanto num ambiente de simulação como num ambiente real, onde diferentes situações foram retratadas. Os resultados obtidos permitem concluir que a solução apresentada é robusta, eficiente para diferentes ambientes, sem que a tarefa de transporte seja comprometida.

Palavras-chave: transporte cooperativo, robôs móveis autónomos, Filtro de Kalman Estendido.

Esta página, intencionalmente, apenas contém esta frase.

Abstract

This dissertation focus on the problem of payload transportation by teams of autonomous mobile robots that act in coordination. The team uses a leader-helper strategy and the control architectures are formalized, recurring to the dynamic attractors approach to behavior based robotics.

On previously developed works there is explicit communication between the leader robot and the helper robot (in the form of heading directions). In this dissertation one develops a module that allows the helper to estimate that information, thus eliminating the need to explicit communication. This contribution will allow the team of robots to move in environments in which communication is restricted, it adds robustness in environments in which the communication is noisy, or it even allows the leader robot to be replaced with a human, in a transparent way from the robot helper's point of view, since it will have the capacity to act independently.

An Extended Kalman Filter was used for the development of the module that estimates the heading direction of the leader robot. As input, this module uses information provided by the measures obtained for the position, orientation and velocity of the leader robot, calculated from the sensorial information gathered from the payload support base in the helper robot.

To validate the solution, experimental tests were conducted, in a simulated environment and in a real environment, where different situations were portrayed. The results obtained in simulated environments allows us to conclude that the presented solution is robust, efficient for different environments, without compromising the transportation task.

Keywords: cooperative transport, autonomous mobile robots, Extended Kalman Filter.

Esta página, intencionalmente, apenas contém esta frase.

Conteúdo

Lista de Figuras	xv
1 Introdução	1
1.1 Robótica móvel	1
1.2 Enquadramento e Objetivos	4
1.3 Estrutura da dissertação	5
2 Fundamentos e Estado da arte	7
2.1 Equipas robô-robô	7
2.2 Interação humano-robô	10
2.3 Estimativa de variáveis	11
2.4 Algoritmo do Filtro de Kalman Estendido	13
3 Robôs Móveis	15
3.1 <i>Hardware</i>	15
3.2 <i>Software</i>	17
3.3 Simulador <i>CoopDynSim</i>	19
4 Arquitetura de controlo	23
4.1 Dinâmica de Navegação	23
4.1.1 Navegação na direção do alvo	25
4.1.2 Evitar obstáculos	27
5 Estimativa da direção de navegação do robô <i>leader</i>	31

5.1	Estimativa baseada em cálculo trigonométrico e vetorial	31
5.2	Filtro de Kalman Estendido	34
5.2.1	Aplicação do Filtro de Kalman Estendido na estimativa da direção de navegação do robô <i>leader</i>	34
5.2.1.1	Etapa de previsão	35
5.2.1.2	Etapa de correção	37
5.3	Implementação do módulo para obter o valor do ângulo que a carga faz com a direção de navegação do robô <i>leader</i>	38
5.3.1	Algoritmo baseado em cálculo trigonométrico e vetorial	38
5.3.2	Algoritmo da implementação do Filtro de Kalman Estendido	40
5.3.3	Cálculo do θ_L	41
6	Resultados	43
6.1	Apresentação dos resultados obtidos	43
6.1.1	Testes realizados no ambiente de simulação CoopDynSim	44
6.1.1.1	Cenário 1	44
6.1.1.2	Cenário 2	47
6.1.1.3	Cenário 3	49
6.1.1.4	Cenário 4	52
6.1.1.5	Cenário 5	54
6.1.1.6	Cenário 6	56
6.1.1.7	Cenário 7	58
6.1.1.8	Cenário 8	60
6.1.2	Testes realizados no ambiente de simulação CoopDynSim utili- zando controlo remoto	62
6.1.2.1	Situação 1	62
6.1.2.2	Situação 2	64
6.1.2.3	Situação 3	65
6.1.2.4	Situação 4	66
6.1.3	Testes realizados no ambiente real	67

6.1.3.1	Interação humano - robô	67
7	Conclusão	71
	Bibliografia	73

Esta página, intencionalmente, apenas contém esta frase.

Lista de Figuras

1.1	Exemplos de robôs móveis.	2
1.2	Sistemas multi-robô.	3
1.3	Equipa de robôs transportando uma carga de grandes dimensões.	4
1.4	Configuração da equipa de robôs.	5
2.1	Sistema robô-escada [2].	8
3.1	Suporte para suportar a carga a transportar.	16
3.2	Um dos robôs reais utilizados nesta dissertação	17
3.3	Esquema do Hardware da Camada de Abstração	18
3.4	Campos do protocolo de comunicação utilizado nas mensagens de rede.	19
3.5	Objetos, Robôs e Alvos	19
4.1	Equipa de robôs e respetivas variáveis.	24
4.2	Representação do alvo e do robô em relação a um referencial externo.	26
4.3	Posições dos sensores de distância.	28
4.4	Obstáculos detetados entre a direção de navegação do robô <i>helper</i> e a carga são movidos para uma posição estratégica, a fim de evitar a colisão da carga com o obstáculo.	28
4.5	Evitar colisões com obstáculos.	30
5.1	$\psi_{c,H}$	32
5.2	Diferença entre a posição do robô <i>leader</i> e a posição do robô <i>helper</i> em ambos os eixos (horizontal e vertical)	33

5.3	Arquitetura do módulo implementado.	38
5.4	$\psi_{c,L}$	41
6.1	Movimento da equipa até atingir o alvo.	45
6.2	θ_L para o cenário 1	47
6.3	Movimento da equipa até atingir o alvo.	48
6.4	θ_L para o cenário 2	49
6.5	Movimento da equipa até atingir o alvo.	50
6.6	θ_L para o cenário 3	51
6.7	θ_L para o cenário 3	52
6.8	Movimento da equipa até atingir o alvo.	53
6.9	θ_L para o cenário 4	54
6.10	Movimento da equipa até atingir o último alvo.	55
6.11	θ_L para o cenário 5	56
6.12	Movimento da equipa até o alvo móvel atingir o alvo fixo.	57
6.13	θ_L para o cenário 6	58
6.14	Movimento da equipa até atingir o alvo móvel.	59
6.15	θ_L para o cenário 7	60
6.16	Movimento da equipa até atingir o último alvo.	61
6.17	θ_L para o cenário 8	62
6.18	Movimento realizado pela equipa na situação 1.	63
6.19	θ_L para a situação 1	63
6.20	Movimento realizado pela equipa na situação 2.	64
6.21	θ_L para a situação 2	64
6.22	Movimento realizado pela equipa na situação 3.	65
6.23	θ_L para a situação 3	65
6.24	Movimento realizado pela equipa na situação 4.	66
6.25	θ_L para a situação 4	67
6.26	Planta do cenário.	68
6.27	Movimento da equipa até chegar à posição de destino.	68
6.28	Trajetória realizada pela equipa humano - robô.	69

6.29 ϕ_{Lest} e θ_{Lest}	70
--	----

Esta página, intencionalmente, apenas contém esta frase.

Capítulo 1

Introdução

Neste capítulo é apresentado o âmbito em que se insere esta dissertação, assim como os aspetos principais que motivaram a sua realização e principais objetivos e resultados que são esperados obter da sua concretização. Por fim, também é apresentada a forma como está organizado este documento.

1.1 Robótica móvel

Os robôs móveis autónomos caracterizam-se por terem a capacidade de se deslocarem autonomamente no ambiente que são inseridos e interagirem com esse ambiente para a realização de tarefas que lhes são confiadas. Para disporem dessas capacidades os robôs são constituídos por um conjunto de *hardware* e de um *software* que lhes proporciona as seguintes habilidades:

- **Locomoção:** possibilita ao robô movimentar-se no ambiente em que é inserido;
- **Sensorização:** permite ao robô medir as suas características e do ambiente que o rodeia;
- **Racionalização:** possibilita ao robô atuar sobre o ambiente que o rodeia;
- **Comunicação:** permite ao robô comunicar com o exterior;

As capacidades apresentadas por estes tipos de robôs permitiu facilmente expandir as suas áreas de atuação (ver Figura 1.1). Por exemplo, na área médica, pode-se mencionar as cadeiras de rodas robotizadas (ver Figura 1.1a). Num ambiente doméstico

foram desenvolvidos aspiradores (ver Figura 1.1b) e cortadores de relva autónomos (ver Figura 1.1c). Na área da indústria foram desenvolvidos robôs que transportam cargas (ver Figura 1.1d).



(a) Cadeira de rodas elétrica Stream da Invacare [9].



(b) Aspirador Roomba 600 da iRobot [10].



(c) Cortador de relva Automower 330x da Husqvarna [8].



(d) AGV da Robotnik [21].

Figura 1.1: Exemplos de robôs móveis.

No sentido de melhorar a eficiência dos robôs móveis na realização de determinadas tarefas, surgem os sistemas multi-robô, sistemas que contribuem de forma coordenada e/ou cooperativa para a realização de tarefas. Os sistemas multi-robô tornaram a realização de determinadas tarefas mais eficiente, rentável e rápida. Um exemplo é no transporte de cargas que permitiu superar algumas limitações que os robôs individuais apresentam na realização dessa tarefa, como por exemplo o robô AGV (Veículo Guiado Automaticamente) que foi dimensionado para desempenhar essa tarefa. Essas limita-

ções devem-se em grande parte ao facto de quando a carga a transportar é de grandes dimensões, o que nestes casos é mais vantajoso a utilização de equipas de robôs, por apresentarem uma maior eficiência e flexibilidade na configuração da equipa, dependendo da carga a transportar.

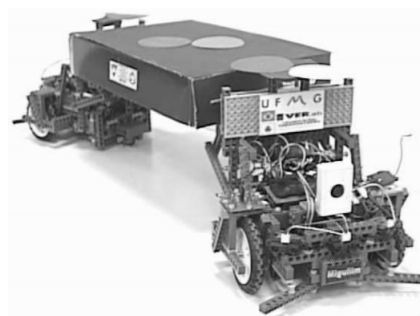
Atendendo às vantagens referidas, estes sistemas têm sido utilizados em inúmeras tarefas (ver Figura 1.2), como por exemplo na exploração de ambientes desconhecidos (ver Figura 1.2a), redes de sensores móveis (ver Figura 1.2b), transporte de objetos de grandes dimensões por equipas multi-robô (ver Figura 1.2c), entre outros.



(a) Exploração de ambientes desconhecidos [3].



(b) Redes de sensores móveis [11].



(c) Transporte de cargas [19].

Figura 1.2: Sistemas multi-robô.

Esta dissertação incidirá na área do transporte de cargas por equipas de dois robôs, onde se pretende tornar o transporte mais eficiente, mais eficaz e mais inteligente.

1.2 Enquadramento e Objetivos

No Departamento de Eletrónica Industrial da Universidade do Minho foram desenvolvidas arquiteturas de controlo que permitem a uma equipa de dois ou mais robôs transportarem cooperativamente cargas de grandes dimensões (ver Figura 1.3), desde uma posição inicial até uma posição destino, evitando obstáculos [23] [14].



Figura 1.3: Equipa de robôs transportando uma carga de grandes dimensões.

Essas arquiteturas foram formalizadas recorrendo à Teoria de Sistemas Dinâmicos (na forma de equações diferenciais não lineares). A equipa é constituída por dois robôs com responsabilidades distintas (Figura 1.4). O *leader* é o único agente que sabe qual o local de destino, informação essa adquirida pelo sistema de visão. Nesse sentido, o robô *leader* tem a responsabilidade de conduzir a equipa de uma posição inicial até uma posição destino evitando os obstáculos. O robô *helper* utiliza a informação comunicada pelo robô *leader*, que neste caso é o ângulo que a carga faz com a direção de navegação do *leader* no momento, para determinar a sua direção de navegação no sentido de orientar a carga de modo que fique alinhada com a direção de navegação do robô *leader*, e ao mesmo tempo evitando obstáculos.

Nos trabalhos anteriores, como foi mencionado, há comunicação explícita de informação entre o robô *leader* e o robô *helper* (na forma de direções de navegação). No presente trabalho pretende-se eliminar a comunicação entre o robô *helper* e o robô *leader*. Sendo assim, o objetivo principal desta dissertação é dotar o robô *helper* de capacidades, que lhe permitam estimar a informação sensorial necessária para que desenvolva autonomamente a tarefa que lhe é proposta. Esta nova contribuição irá permitir

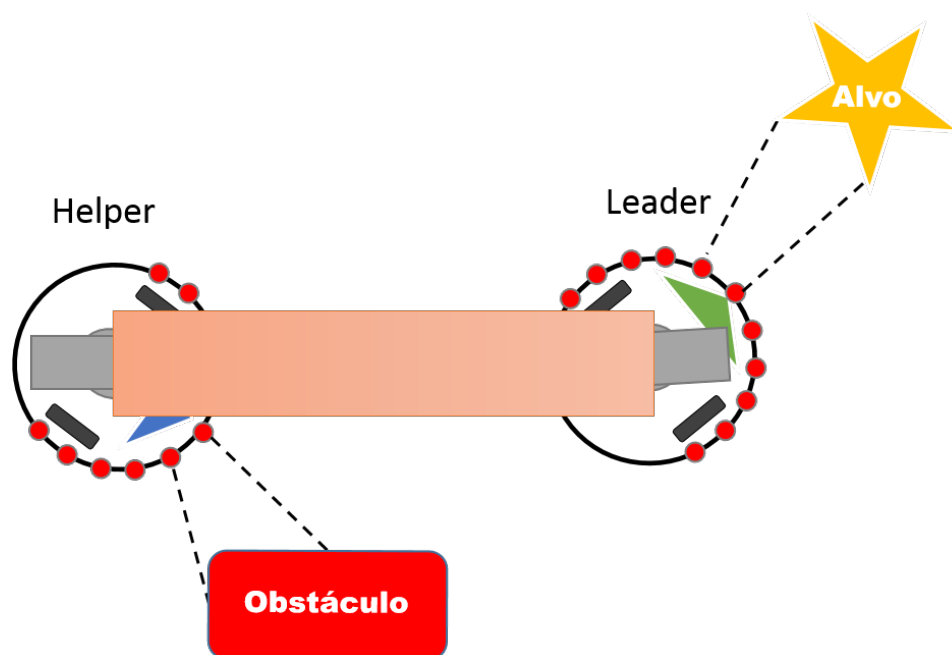


Figura 1.4: Configuração da equipa de robôs.

que a equipa de robôs possa movimentar-se em ambientes em que a comunicação não é permitida, adiciona robustez em ambientes em que a comunicação é ruidosa ou, ainda, permite que o robô *leader* possa ser substituído por um humano, de forma transparente do ponto de vista do robô *helper*, uma vez que terá capacidades para agir de forma independente. A existência desta cooperação humano-robô na tarefa de transporte permitirá a realização da mesma, de uma forma mais fiável em ambientes mais complexos, pela contribuição do humano a nível de inteligência e de experiência no conhecimento global da tarefa.

1.3 Estrutura da dissertação

Este documento está dividido em 7 capítulos.

No capítulo 1 é apresentada a introdução ao tema, a motivação para a realização desta dissertação e ainda os objetivos a realizar nesta dissertação.

No capítulo 2 é feita uma breve análise de projetos, no âmbito do transporte de cargas entre equipas de dois robôs e equipas humano - robô e ainda projetos que utilizam

métodos de estimativas de parâmetros na interação entre robôs.

No capítulo 3 é apresentada a equipa de robôs a nível de *software* e *hardware*, utilizada nesta dissertação.

A arquitetura de controlo que coordena o movimento da equipa para o alvo e ao mesmo tempo evite obstáculos é apresentada no capítulo 4.

O módulo implementado, que fará com que o robô *helper* tenha a capacidade de estimar a direção de navegação do robô *leader*, é apresentado no capítulo 5.

No capítulo 6 são apresentados os principais resultados obtidos pela implementação do algoritmo tanto num ambiente de simulação como num ambiente real. E ainda uma análise aos resultados obtidos.

As principais conclusões desta dissertação e perspetivas quanto ao trabalho futuro são apresentadas no capítulo 7.

Capítulo 2

Fundamentos e Estado da arte

Neste capítulo são apresentados alguns projetos desenvolvidos no âmbito do transporte de cargas entre equipas de dois robôs e equipas humano - robô e ainda projetos que utilizam métodos de estimativas de parâmetros na interação entre robôs. Para cada projeto apresentado é feita uma análise, relativamente às principais características que caracterizam cada projeto e assim perspetivando o melhor método a aplicar tendo em conta o objetivo desta dissertação.

2.1 Equipas robô-robô

A realização de determinadas tarefas requer a cooperação de dois ou mais agentes. Um exemplo é na tarefa de transporte. Atualmente, tem sido cada vez mais os projetos de investigação que utilizam equipas de robôs no transporte de cargas devido a uma maior eficiência no caso da carga a transportar ser de grandes dimensões. Nesse sentido, têm-se desenvolvido várias abordagens que permitem a uma equipa de robôs cooperar entre si na realização da tarefa, como é o caso dos dois projetos apresentados de seguida.

No trabalho *"Distributed Ladder Transportation Algorithm for Two Robots in a Corridor"* [2] é abordado o problema do transporte de objetos longos, como uma escada ou qualquer outro objeto longo, num corredor com um canto de 90 graus utilizando uma equipa de dois robôs omnidireccionais. Para este problema houve muitas variações: os robôs podiam possuir as mesmas características ou não e em relação ao espaço de

trabalho poderia haver ou não a presença de obstáculos. Tendo em consideração essas variações do problema a abordagem que foi considerada como mais vantajosa foi uma abordagem distribuída onde cada robô planeia o seu movimento com base nas posições atuais e futuras da escada, da localização das paredes e do movimento do outro robô, observado indiretamente através da ligação entre o robô e a escada. Nesse sentido, não existe qualquer comunicação explícita entre os robôs.

Na abordagem implementada pelos autores cada robô individualmente calcula, sem ter em conta a coordenação do movimento, o caminho a seguir entre a posição inicial e final ao longo do canto, para depois seguirem-na durante o movimento. Em seguida, os robôs iniciam o seu movimento executando repetidamente cada passo até que ambos os robôs cheguem à posição de destino.

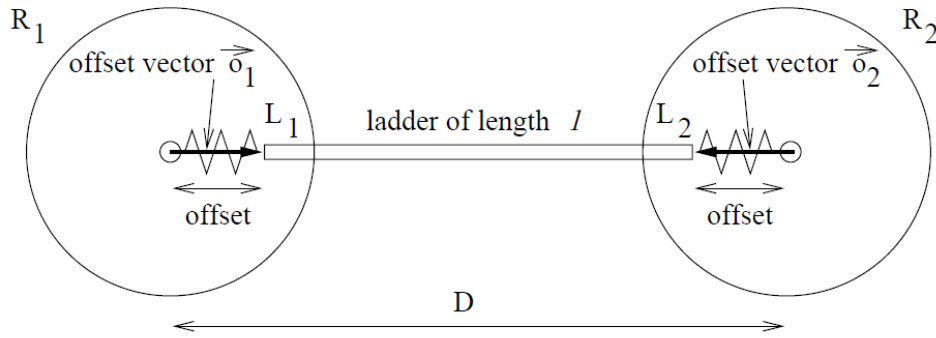


Figura 2.1: Sistema robô-escada [2].

Na Figura 2.1 está representado o sistema robô-escada, onde o objeto a ser transportado, a escada, é representado pelo segmento de linha $\overline{L_1L_2}$, e os dois robôs designados por R_1 e R_2 . As extremidades da escada são unidas aos dois robôs por meio de um sensor de força flexível, onde cada sensor é modelado como uma mola ideal de constante sc_i . O comprimento da escada é representado por l e ainda o comprimento entre os centros dos robôs representado por D . Sempre que exista diferença entre D e l durante o movimento existe *offset* entre R_i e L_i , que é representado pelo vetor *offset* \vec{o}_i . Para alcançar o equilíbrio a força exercida pelas duas molas, que unem a extremidade da escada ao centro de cada robô, deve ser a mesma.

Nos testes que foram realizados os autores concluíram que o algoritmo é robusto

nas diferenças existentes nas características dos robôs, exceto na velocidade máxima e na constante da mola que devem ser as mesmas em ambos os robôs.

No trabalho "*Coordination of multiple mobile robots in a object carrying task using implicit communication*" [19] os autores abordam o problema da coordenação e controle de uma equipa de robôs no transporte de uma caixa. A equipa é composta por um líder e por um seguidor, onde o líder comanda a equipa para um alvo pré-definido e é o que tem conhecimento da posição do alvo. Ambos os robôs possuem um suporte para o objeto que é sensível à força. Enquanto o robô líder navega numa determinada direção o robô seguidor tenta manter a força, aplicada ao objeto, constante. Inicialmente, o objeto está em equilíbrio desde que o somatório das forças aplicadas ao objeto seja igual a zero.

Para facilitar a tarefa de evitar obstáculos a liderança da equipa é mudada ao longo da navegação por meio de comunicação implícita. Essa comunicação entre a equipa é realizada através da interação de cada robô com o objeto. Essa mudança de liderança é feita sempre que o seguidor detete que o líder recua sem que haja comunicação explícita.

Nos testes realizados os autores concluíram que a utilização da comunicação implícita se torna eficaz em casos que a informação trocada, entre os robôs, é simples. Para outras situações onde os dados a transmitir são mais complexos, este tipo de comunicação é mais difícil ou mesmo impossível de implementar.

Os trabalhos mencionados, nesta secção, apresentam semelhanças com o trabalho desenvolvido nesta dissertação, nomeadamente, a nível da comunicação entre a equipa de robôs, em que entre eles, não existe comunicação explícita.

No trabalho [2] os autores apresentam uma abordagem que permite utilizar robôs com diferentes características. Contudo, a velocidade máxima e a constante da mola dos dois robôs tem de ser a mesma para que a tarefa de transporte não seja comprometida.

No trabalho [19] é apresentada uma abordagem que permite uma maior facilidade na tarefa de evitar obstáculos, por parte da equipa de robôs, através da mudança de liderança da equipa por meio de comunicação implícita. Contudo, a utilização da comunicação implícita só é possível quando a informação a ser transmitida entre a equipa é simples, limitando assim a sua utilização a tarefas específicas.

Uma das vantagens do trabalho da presente dissertação em relação aos trabalhos mencionados, nesta secção, é a possibilidade do robô líder poder ser substituído por um humano, tornando assim a realização da tarefa de transporte mais fiável.

2.2 Interação humano-robô

A existência da interação entre humanos e robôs na realização de determinadas tarefas tem sido uma mais valia, devido ao facto de o humano ter um maior conhecimento sobre a tarefa a realizar, tornando a sua realização mais fiável, nomeadamente em ambientes mais complexos.

No âmbito do transporte de cargas têm-se desenvolvido várias metodologias que permitem a interação entre robôs e humanos na realização da tarefa, como é o caso das duas metodologias apresentadas de seguida.

No trabalho *"The role of roles: Physical cooperation between humans and robots"* [18] é apresentada uma análise sistemática da interação humano – robô e a introdução de três diferentes estratégias para a atribuição do esforço na realização da tarefa do transporte de uma mesa. Estas estratégias têm por base a repartição do esforço entre o humano e o robô na realização da tarefa. A primeira estratégia resulta numa alocação de papel constante (CRA) onde o robô contribui para a tarefa sem mudança de papel, ou seja, não é tido em conta a vontade do humano em acelerar ou desacelerar no progresso da tarefa. A segunda estratégia deriva de uma alocação de papel ponderada (WPRA) que resulta num ajuste contínuo dos papéis atribuídos, já tendo em consideração o feedback do ser humano. Por fim a última estratégia resulta numa alocação de papel ponderada discreta (DPRA), semelhante à segunda estratégia mas de modo discreto, ou seja, a atribuição do papel advém de um pequeno número de passos significativos distintos.

Os testes realizados revelaram que uma política de atribuição de um papel dinâmico contínuo é objetivamente superior sobre uma estratégia de papel constante ao longo da realização da tarefa de transporte.

No trabalho *"Feedback Motion Planning and Learning from Demonstration in Phy-*

sical Robotic Assistance: Differences and Synergies" [12] é desenvolvida uma nova abordagem com o objetivo de facilitar o comportamento na interação física entre um robô e um humano. Essa abordagem é baseada na combinação conjunta de estratégias baseadas em planeamento e em aprendizagem.

Na abordagem baseada em planeamento o robô calcula o seu plano de movimento tendo em conta as restrições do espaço envolvente. Em consequência, o plano de movimento do robô geralmente difere do plano de movimento do humano, o que implica ajuste contínuo do plano de movimento do robô.

Na abordagem baseada em aprendizagem o robô comporta-se de forma passiva durante a execução da tarefa e observa a trajetória executada que pode ser assumida como sendo próxima ao do parceiro humano.

Nos resultados obtidos os autores concluíram que a combinação destas abordagens baseadas em planeamento e em aprendizagem permitiram um desempenho superior do que a utilização destas abordagens de forma individual.

2.3 Estimativa de variáveis

Na interação entre equipas robô - robô ou humano - robô pode ser necessário adquirir informação do líder por parte dos seus ajudantes, ao longo da realização da tarefa. Essa informação pode ser adquirida através de comunicação explícita ou de forma implícita. No caso da comunicação explícita pode ser entendida como uma ação específica projetada unicamente para transmitir informações para outros robôs da equipa. Por exemplo na navegação líder-seguidor é necessário que o líder comunique a sua posição ao seu seguidor. Em relação à comunicação implícita é definida como um efeito colateral das ações dos robôs ou através da forma como eles atuam sobre o ambiente em redor. Um exemplo é no transporte de cargas que a troca de informação entre o líder e o ajudante é realizada através da interação de cada robô com o objeto a transportar.

Tendo em conta os objetivos desta dissertação, que é a eliminação da comunicação explícita existente entre a equipa de robôs, nesta secção são apresentados projetos que utilizam métodos de estimativas de variáveis para obterem informação do líder, em vez

de utilizarem comunicação explícita entre a equipa de robôs.

No trabalho “*Vision-based Leader-Follower with Limited Information*” [17] é apresentado um novo algoritmo para a navegação líder-seguidor onde a trajetória do líder é desconhecida para os seus seguidores. No algoritmo abordado pelos autores é considerada a navegação em linha reta e na diagonal e não requer comunicação entre robôs. Os autores utilizaram o Extended Kalman Filter (EKF) tendo como entrada a medição da imagem local capturada através da câmara, que se encontra montada em cada robô, para estimar a posição do líder por parte dos seguidores. No algoritmo apresentado, pelos autores neste trabalho, existem vários aspetos que o tornam vantajoso em relação a outras abordagens semelhantes. A utilização da câmara sem depender da utilização de marcadores para o reconhecimento do líder, o uso apenas da medição da imagem para cada seguidor estimar a posição do líder e um controlo descentralizado que permite que cada membro da equipa possa ser um líder, um seguidor ou ambos.

No trabalho “*Leader – follower formation control without leader’s velocity information*” [22] é considerado o problema do controlo da formação de robôs móveis onde não se tem medição direta da velocidade linear do robô líder por parte dos seus seguidores. Os autores propuseram dois algoritmos não lineares descentralizados, um deles foi o controlo adaptativo com feedback dinâmico e o outro método foi o *immersion & invariance*. Segundo os autores o método *immersion & invariance* revelou uma melhor performance na estimativa da velocidade linear do líder do que o controlo adaptativo com feedback dinâmico, sendo um dos motivos devido ao facto deste último não considerar qualquer perturbação no sistema. A performance dos controladores a utilizar depende largamente da precisão do modelo e é vulnerável a incertezas. O método *immersion & invariance* permite superar essa vulnerabilidade sendo mais robusto na estimativa da velocidade linear do líder, mas mais complexo de se aplicar do que o controlo adaptativo com feedback dinâmico que é mais simples de dedução.

2.4 Algoritmo do Filtro de Kalman Estendido

Uma das técnicas que tem sido muito utilizada para a estimativa de dados baseia-se na utilização das diversas variantes dos Filtros de Kalman. O filtro utiliza um modelo dinâmico de um sistema, entradas de controlo conhecidas e medições para gerar uma estimativa das variáveis do sistema.

Contudo, o modelo que descreve o comportamento do robô *leader*, ao longo da navegação, é um sistema não linear o que impossibilita a utilização do Filtro de Kalman, visto que este só se aplica a sistemas lineares. Neste caso, usa-se uma variante do filtro, designado Filtro de Kalman Estendido, que utiliza um processo de linearização, para cada instante de tempo, o que permite uma aproximação do sistema não linear a um sistema linear.

O Filtro de Kalman Estendido [5] na sua implementação utiliza duas etapas, a de previsão e a de correção, para a obtenção da estimativa do estado.

Na etapa de previsão são previstas as estimativas, de estado e da covariância do erro atuais, para obter as estimativas, *a priori*, para o próximo instante de tempo.

Na previsão do estado (2.2) a matriz de transição, Φ_k , é uma expressão não linear do modelo.

Na previsão do erro de covariância (2.3) esta será modificada pela forma linearizada do modelo (2.1), obtida através do Jacobiano, e pela adição da matriz de covariância do estado, Q_k , transformada pela matriz Jacobiana, Γ_k .

$$F_k^- = \frac{\partial f}{\partial x}(\hat{x}_k^-) \quad (2.1)$$

$$X_k^- = \Phi_k X_{k-1} \quad (2.2)$$

$$P_k^- = F_k P_{k-1} F_k^T + \Gamma_k Q_k \Gamma_k^T \quad (2.3)$$

Na etapa de correção o estado é corrigido pelo modelo de observação, de modo que a covariância do valor estimado seja minimizada. O ganho de Kalman (2.4) é uma função da estimativa do erro, P_k^- , da matriz de medição, H_k , que é linearizada pela

matriz Jacobiana (2.5), e da matriz de covariância da medição do sensor, R_k . O ganho de Kalman é utilizado para corrigir a estimativa do estado (2.6) e reduzir a covariância de estado (2.7).

$$K_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (2.4)$$

$$H_k^- = \frac{\partial h}{\partial x}(\hat{x}_k^-) \quad (2.5)$$

$$X_k = X_k^- + K_k [Z_k - h X_k^-] \quad (2.6)$$

$$P_k = [I - K_k H_k] P_k^- \quad (2.7)$$

A matriz de covariância, P_k , permite que o filtro evolua ao longo do tempo, visto que fará com que o filtro dê mais credibilidade às medições ou às estimativas. No caso da matriz de covariância, P_k , tender para zero, o ganho de Kalman também tenderá para zero, o que faz com que o filtro pese mais as previsões que os resíduos, ou seja, o filtro confiará mais na dinâmica do que nas medições. Para o caso da matriz de covariância, P_k , ter valores superiores (distanciados do zero) o ganho de Kalman tenderá a aumentar, o que faz com que o filtro pese mais os resíduos do que as previsões, ou seja, o filtro confiará mais nas medições do que na dinâmica.

Contudo, o Filtro de Kalman Estendido não garante teoricamente a convergência nem que as suas estimativas sejam ótimas. Porém, na maioria das vezes, consegue um desempenho muito satisfatório. Para além disso, evidencia uma complexidade na sintonização dos seus parâmetros, nomeadamente, na matriz de covariância da medição do sensor, R_k , e na matriz de covariância do estado, Q_k .

No entanto este tipo de filtro tem sido muito utilizado na robótica, nomeadamente, na estimativa da localização do robô [7], na deteção de objetos [24], no caminho a seguir [1], entre outros.

Capítulo 3

Robôs Móveis

Neste capítulo é apresentada a constituição da equipa de robôs, utilizada nesta dissertação, em termos de *hardware* e de *software*. E por fim, faz-se uma breve descrição do simulador que servirá de base à validação dos algoritmos desenvolvidos no âmbito desta dissertação.

3.1 *Hardware*

A equipa de robôs é constituída por dois robôs móveis autónomos idênticos entre si, tanto na forma como nas características de *hardware* (ver Figura 3.2). Cada robô possui um conjunto de *hardware*, que é descrito de seguida:

- **Uma bússola** tem como finalidade obter a direção de navegação do robô (ϕ_r), em relação a um referencial externo;
- **Um sistema de visão omnidirecional** para permitir ao robô *leader* conduzir a equipa para posições desejadas (alvos);
- **Duas rodas motrizes diferenciais**;
- **Onze sensores de distância (tipo infravermelho)** para medir distâncias e detetar possíveis obstáculos nas direções que apontam;
- **Um router sem fios** para permitir acesso remoto a fim de facilitar a configuração dos robôs;
- **Seis baterias de chumbo-ácido de 12 V com 7.2 Ah de capacidade de energia**

para fornecer energia a todo o *hardware*;

- **Um suporte dedicado para suportar a carga a transportar** e fornecer informação do movimento desta (Figura 3.1). O suporte é constituído por uma junta rotacional passiva e uma junta prismática semi-passiva (atuação por molas para a retornar para o centro) (Figura 3.1a). A informação que é possível retirar do suporte é a distância da carga em relação ao centro do robô, $d_{c,r}$, em que a distância permitida varia entre $-D_{c,min} = D_{c,max} = 200\text{ mm}$. Adicionalmente é possível obter a direção da carga, $\psi_{c,r}$, relativamente à direção de navegação do robô, θ_r . Como ilustrado na Figura 3.1b a orientação da carga relativamente ao referencial externo é dada por: $\psi_{c,r} = \phi_r + \theta_r$;

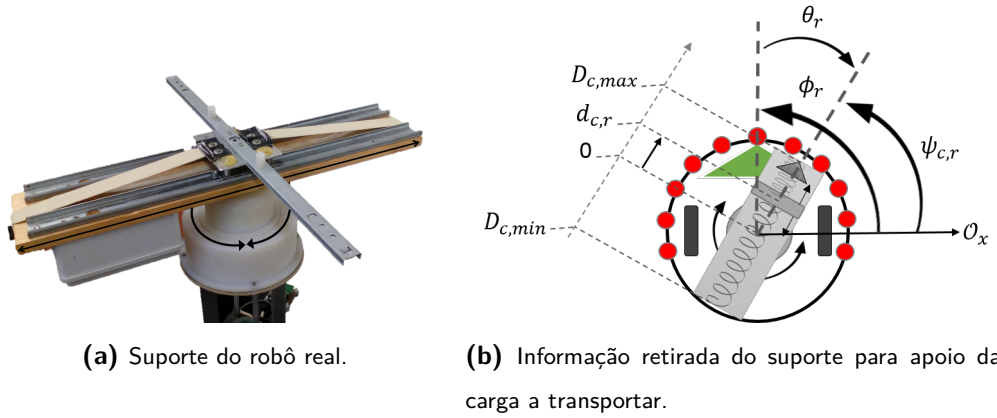


Figura 3.1: Suporte para suportar a carga a transportar.

- Um *embedded PC* com um processador *VIA* de um núcleo a 1.0 GHz, 512 MB de RAM e com o sistema operativo *Windows Embedded Standard 7*;



Figura 3.2: Um dos robôs reais utilizados nesta dissertação

3.2 Software

Na arquitetura desenvolvida cada robô móvel tem a sua aplicação de controlo, a qual é responsável pelo processamento de toda a informação sensorial e respetiva atuação motora. Para que a aplicação de controlo de cada robô aceda aos componentes de *hardware* (sensores e atuadores) é necessário estabelecer comunicação entre ambos. Sendo assim, foi desenvolvida uma camada de *middleware*, baseada no YARP, para cada um dos componentes de *hardware* [13]. O YARP (Yet Another Robot Platform) [16] é uma biblioteca *open source* que é utilizada no desenvolvimento de *software* para aplicações robóticas em tempo real e que envolvem interface com *hardware* diverso (sensores e atuadores) e em mudança.

A camada de *middleware* desenvolvida é o que permite à aplicação de controlo o acesso à informação sensorial e a atuação motora através de uma tipologia cliente - servidor, onde as aplicações de controlo dos robôs são clientes e os *middlewares* são servidores. Na Figura 3.3 estão apresentadas as diversas camadas que permitem à aplicação de controlo aceder ao *hardware*. A comunicação entre a aplicação de controlo

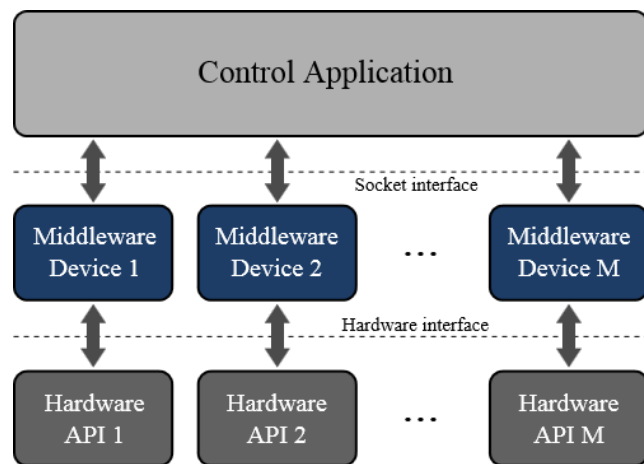


Figura 3.3: Esquema do Hardware da Camada de Abstração

e a camada de *middleware* é realizada através de *sockets* e a comunicação entre a camada de *middleware* e o respetivo *hardware* é realizada através das respetivas API's (Application Program Interface) do *hardware*.

As mensagens trocadas entre a aplicação de controlo e os respetivos *middlewares* seguem o protocolo de comunicação TCP/IP (Transmission Control Protocol/Internet Protocol) (ver Figura 3.4) sendo composta pelos seguintes campos:

- Error Code: número inteiro para sinalizar a ocorrência de um erro;
- Text: mensagem de texto (poderá ser uma descrição do erro);
- Command: número inteiro positivo constante da lista de comandos;
- P Parameters: número inteiro para indicar a quantidade, P , de *Parameters* que a mensagem contém;
- Parameters: vetor de dimensão P com dados inteiros;
- D Data: número inteiro para indicar a quantidade, D , de *Data* que a mensagem contém;
- Data: vetor de dimensão D com dados flutuantes de precisão simples.

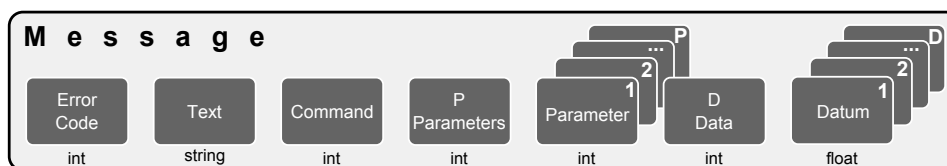


Figura 3.4: Campos do protocolo de comunicação utilizado nas mensagens de rede.

3.3 Simulador *CoopDynSim*

O ambiente de simulação que serviu de apoio para a validação da metodologia implementada foi o *CoopDynSim* [13], um simulador desenvolvido no Laboratório de Robótica Antropomórfica e Móvel da Universidade do Minho. O *CoopDynSim* é um simulador robótico 3D, desenvolvido em C++, capaz de emular múltiplos robôs ou equipas de robôs, obstáculos e alvos baseado em Newton Game Dynamics, Open Graphics Library e YARP.

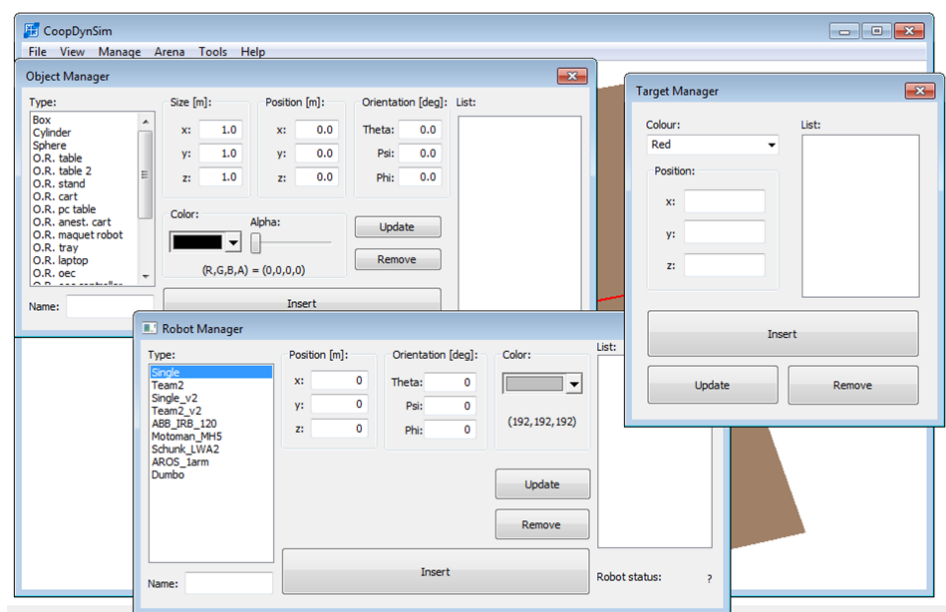


Figura 3.5: Objetos, Robôs e Alvos

Na interface gráfica do simulador é possível aceder através do Manager menu aos três constituintes básicos do ambiente virtual (ver Figura 3.5): objetos, robôs e alvos. Na componente **objetos** é possível inserir no ambiente virtual uma caixa, um cilindro e/ou uma esfera, onde é possível especificar o tamanho, posição e a orientação em relação

ao mundo virtual. Na componente **robôs** é possível optar entre robôs individuais ou equipas de robôs, onde equipas de robôs serão posteriormente utilizadas para tarefas de transporte. Também é possível especificar a posição e a orientação dos robôs e ainda a mudança da cor e do nome dos robôs para a diferenciação dos robôs. O nome permite identificar a plataforma virtual na rede, logo a necessidade de terem nomes diferentes. Estes robôs têm as mesmas características e interface como os robôs reais (dimensões, sensores e atuadores) e seguem uma arquitetura cliente - servidor, onde cada robô é composto por vários módulos de *hardware* que atuam como servidores e a aplicação de controlo tem clientes que se conetam a cada um desses módulos. Por fim, a componente **alvos** que representa um tipo especial de objetos (marcos coloridos) que especificam destinos desejados para os robôs alcançarem. Cada plataforma móvel tem um módulo de destino que retorna a distância e o deslocamento angular para estes marcadores, simulando assim, o sistema de visão.

A fim de diminuir o esforço necessário para configurar uma experiência também estão disponíveis elementos mais complexos, tais como mundos e cenários. Os elementos mundos são compostos por um piso e vários objetos, onde é possível definir as dimensões do piso (comprimento, largura e altura) e as propriedades de cada objeto (tipo, tamanho, posição, orientação e massa). Em relação aos cenários já se encontram disponíveis para serem descarregados e prontos a serem utilizados em simulação.

As principais características que este simulador apresenta reside na modularidade e no nível de abstração dos componentes robóticos, através da utilização da camada *middleware*. Isto deve-se ao facto da camada de *middleware* utilizada ser baseada no YARP que providencia um encapsulamento com uma interface baseada em *sockets* para cada módulo de *hardware*. Uma vez que esta camada de abstração é empregada tanto na simulação como na implementação real, a mesma aplicação de controlo pode ser utilizada em ambas as situações, eliminando assim, na maioria das vezes, a migração da simulação para as plataformas reais, o que permite diminuir o esforço e o tempo gasto na migração da simulação para a implementação real. No presente trabalho o módulo implementado foi desenvolvido em Matlab e facilmente foi introduzido no robô real. Esta característica do simulador permitiu uma compilação direta do executável gerado

pelo Matlab no robô real. Para além disso, este simulador ainda dispõe da capacidade de correr em tempo real ou em tempo simulado o que possibilita uma maior flexibilidade do *software*, e ainda permite ao utilizador a reprodução da simulação.

Esta página, intencionalmente, apenas contém esta frase.

Capítulo 4

Arquitetura de controlo

Neste capítulo é apresentada a arquitetura de controlo que permite coordenar o movimento da equipa de robôs para a posição desejada e evitando obstáculos ao mesmo tempo. Estas arquiteturas foram desenvolvidas em trabalhos anteriores e serão estendidas nesta dissertação [14], [4].

4.1 Dinâmica de Navegação

Na equipa, cada robô desempenha uma função diferente, um *leader* outro *helper*, durante a realização da tarefa de transporte. Nesse sentido, o controlo e a coordenação da equipa assenta no seguinte:

- O robô *leader* é o único que tem conhecimento da posição de destino. Para tal utiliza um sistema de visão que adquire a informação dos alvos, de forma a navegar até ao local de destino. Enquanto a equipa navega o robô *leader* comunica o valor do ângulo que a carga faz com a sua direção de navegação (θ_L) ao robô *helper* (ver Figura 4.1);
- O robô *helper* utiliza essa informação conjuntamente com o valor do ângulo que a carga faz com a sua direção de navegação no momento (θ_H) e com a distância da carga em relação ao seu centro (d_H), para determinar a sua direção de navegação (ϕ_H) e a sua velocidade angular e linear. Em relação à direção de navegação do robô *helper*, o que se pretende nesta arquitetura de controlo é que o robô

helper oriente a carga para que fique alinhada com a direção de navegação do robô *leader* (ϕ_L) e ao mesmo tempo evite os obstáculos;

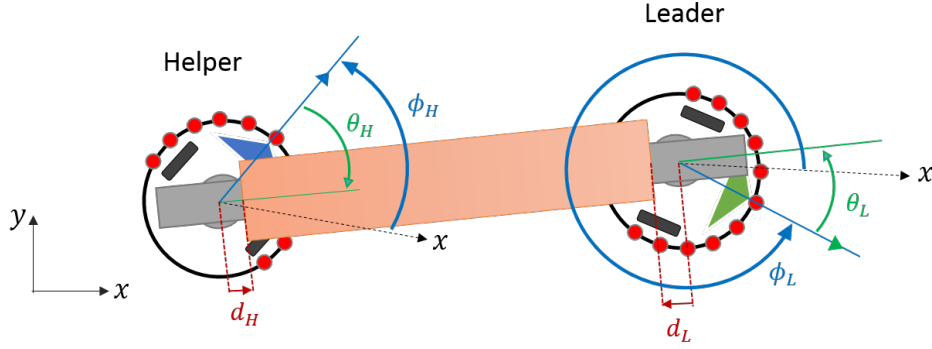


Figura 4.1: Equipe de robôs. Direção de navegação do *leader*, ϕ_L , direção de navegação do *helper*, ϕ_H . Distância da carga ao centro do robô, d_L ou d_H . Ângulo que a carga faz com a direção corrente do robô *leader* θ_L e do *helper* θ_H .

Para que isto seja possível e a tarefa de transporte seja bem sucedida as arquiteturas de controlo desenvolvidas e que serão estendidas nesta dissertação foram formalizadas recorrendo à Teoria de Sistemas Dinâmicos (na forma de equações diferenciais não lineares)[14]. Nessas arquiteturas a dinâmica comportamental da direção de navegação, $\phi_r(t)$, e da velocidade linear, $v_r(t)$, de cada robô r são regidas pelas seguintes equações diferenciais:

$$\frac{d\phi_r(t)}{dt} = f_{des,r}(\phi_r(t)) + f_{obs,r}(\phi_r(t)) + f_{stoch,r} \quad (4.1)$$

$$\frac{dv_r(t)}{dt} = -\lambda_{v,r}(v_r(t) - v_{des,r}), \lambda_{v,r} > 0 \quad (4.2)$$

Em (4.1) está presente a dinâmica comportamental da direção de navegação do robô r , que é modelada pelas componentes: $f_{des,r}(\phi_r(t))$, que representa o movimento do robô na direção do alvo; $f_{obs,r}(\phi_r(t))$, que representa o comportamento “evitar obstáculos”; e $f_{stoch,r}$, que representa uma força estocástica que garante a “fuga” a estados de equilíbrio instáveis. Esta força estocástica (4.3) consiste em ruído branco Gaussiano, ξ_n , de variância unitária, pelo que ϑ é a variância efetiva da força estocástica.

$$f_{stoch} = \sqrt{\vartheta}\xi_n \quad (4.3)$$

Em (4.2) está representado um sistema dinâmico linear que atrai a velocidade linear do robô r para o valor desejado $v_{des,r}$ com uma taxa de relaxação definida por $(\lambda_{v,r})$.

4.1.1 Navegação na direção do alvo

A componente $f_{des,H}(\phi_r(t))(4.4)$, no caso do robô *helper*, é responsável por orientar o robô *helper* na direção desejada e permitir rapidamente alinhar a carga com a direção de navegação do robô *leader*, sendo definida por:

$$f_{des,H}(\phi_H(t)) = \lambda_{trp,H} \text{sen}(\theta_H + \gamma_{\theta,L}) \quad (4.4)$$

Nesta componente é tido em conta o ângulo que a carga faz com a direção atual do robô *helper*, θ_H , e é evidenciada a importância de adquirir informação do robô *leader* por parte do robô *helper* pela variável $\gamma_{\theta,L}$ (4.5), na medida em que esta componente contribui para o alinhamento da carga com a direção de navegação do robô *leader*, sendo definida por:

$$\gamma_{\theta,L} = \begin{cases} \gamma_{fac} \lambda_{\theta_L} \left(\frac{2}{1 + \exp(\beta_{\theta_L} \theta_L)} - 1 \right) & , \Delta L > 0, \\ \theta_L & , otherwise. \end{cases} \quad (4.5)$$

A variável λ_{θ_L} limita o ângulo de curvatura do *helper* (que neste caso é $\pi/2$); θ_L é o ângulo que a carga faz com a direção atual do *leader*; β_{θ_L} é o parâmetro usado para ajustar o raio de curvatura desejado para o *helper* alinhar a carga com a direção do *leader*; γ_{fac} é usado para reduzir o valor do raio da curvatura desejada para que o *helper* não deixe cair a carga, dando prioridade a recuperar o deslocamento da carga sobre o suporte, relativamente à tarefa de alinhar a carga com a direção atual do *leader*. No caso da variável γ_{fac} (4.6) o seu valor é obtido tendo em conta a distância da carga em relação ao centro do robô *helper* no instante atual, d_H , e o valor máximo permitido para essa distância (20 cm), $d_{H,max}$. O parâmetro n é utilizado para ajustar a taxa de decaimento de γ_{fac} .

$$\gamma_{fac} = 1 - |d_H/d_{H,max}|^n \quad (4.6)$$

No caso do parâmetro ΔL , que determina o valor da variável $\gamma_{\theta,L}$, o seu valor (4.7) está dependente da direção de navegação do *leader*, ou seja, se estiver a navegar numa direção que tende a aumentar a distância entre os robôs $\Delta L = 1$ se não $\Delta L = -1$.

$$\Delta L = \begin{cases} 1 & , -\pi/2 \leq \theta_L < \pi/2, \\ -1 & , otherwise. \end{cases} , \forall \theta_H \in [-\pi, \pi] \quad (4.7)$$

Nesta componente, como foi mencionado anteriormente, é necessário que o robô *helper* adquira informação sobre o robô *leader*, que neste caso é o ângulo que a carga faz com a direção de navegação do robô *leader*. Como se disse no início do capítulo. Nos trabalhos desenvolvidos anteriormente essa informação era comunicada pelo robô *leader*. Nesta dissertação essa informação será estimada, eliminando assim qualquer comunicação entre o *leader* e o *helper*.

No caso do robô *leader*, a componente $f_{des,L}(\phi)$ (4.8) é responsável por orientar o robô *leader* na direção dos alvos, que correspondem a posições desejadas ao longo da navegação, sendo definida por [4]:

$$f_{des,L}(\phi_L) = -\lambda_{alvo} \text{sen}(\phi_L - \psi_{alvo}) \quad (4.8)$$

Nesta componente a variável λ_{alvo} controla a intensidade da força com que o ângulo do robô, ϕ_L , se direciona para a direção do alvo. Essa direção (ver Figura 4.2), ψ_{alvo} , em relação ao referencial externo, que o robô *leader* “vê” o alvo a partir da sua posição atual é definida por:

$$\psi_{alvo} = \arctan \left(\frac{y_{alvo} - y_{robô}}{x_{alvo} - x_{robô}} \right)$$

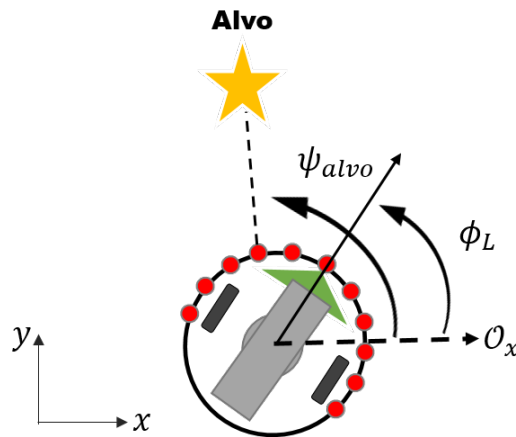


Figura 4.2: Representação do alvo e do robô em relação a um referencial externo.

4.1.2 Evitar obstáculos

O comportamento que permite ao robô *helper* “evitar obstáculos” resulta do somatório das contribuições de cada sensor de distância na presença de obstáculos (4.9).

$$f_{obs,H}(\phi_H(t)) = \sum_{i=1}^{nSectors} f_{obs,i,H}(\phi_H(t)) \quad (4.9)$$

Onde $nSectors$ é o número de setores de sensores de distâncias que o robô possui, e $f_{obs,H}(\phi_H(t))$ representa a contribuição do sensor de distância no setor i . Cada contribuição é definida por:

$$f_{obs,H}(\phi_H(t)) = \lambda_{obs,i}(\phi_H(t) - \psi_{obs,i}) * \exp\left(-\frac{(\phi_H(t) - \psi_{obs,i})^2}{2\sigma_i^2}\right) \quad (4.10)$$

Na equação (4.10) $\psi_{obs,i}$ representa a direção onde se encontra o obstáculo, com uma força de repulsão definida por $\lambda_{obs,i} > 0$ e com σ_i definindo a gama angular sobre a qual o repulsor exerce sua força repulsiva.

Em princípio quando o robô deteta um obstáculo numa determinada direção deve-se afastar dessa direção. Mas devido ao facto da equipa estar ligada por um objeto que transporta cooperativamente a sua presença deve ser considerada na dinâmica comportamental. Nesse sentido, a direção onde é detetado o obstáculo por parte de cada sensor (4.11), $\psi_{obs,i}$, é definida tendo em conta a direção de navegação do robô, ϕ_H , e o parâmetro $\Psi_{obs,i}$ que considera a existência do objeto:

$$\psi_{obs,i} = \phi_H + \Psi_{obs,i} \quad (4.11)$$

Relativamente ao valor do parâmetro, $\Psi_{obs,i}$, (4.12) é definido tendo em conta o ângulo que a carga faz com a direção atual do robô *helper*, θ_H .

$$\Psi_{obs,i} = \begin{cases} -\delta & , (\theta_H \geq 0) \wedge (0 \leq \varrho_i \delta \leq \theta_H) \\ +\delta & , (\theta_H < 0) \wedge (\theta_H \leq \varrho_i \delta \leq 0) \\ \varrho_i \delta & , otherwise \end{cases} \quad (4.12)$$

$$, \forall \theta_H \in [-\pi, \pi] \wedge \forall i \in \mathbb{N} : i \leq nSectors$$

Onde δ representa a distância angular entre o centro de dois sensores de distância consecutivos (ver Figura 4.3) e $\varrho_i = \{-5, -4, \dots, 4, 5\}$ é a posição setor em relação à direção atual do robô.

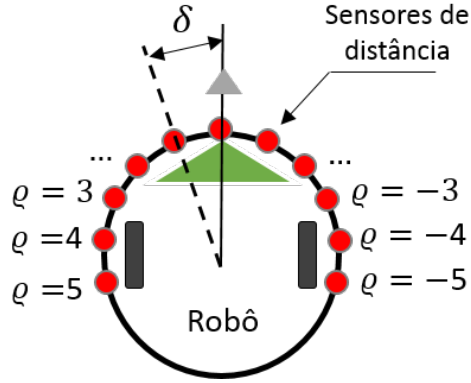


Figura 4.3: Posições dos sensores de distância.

Na Figura 4.4 é retratada uma situação que demonstra a importância desta abordagem no cálculo da direção que é detectado o obstáculo. Neste caso o robô *helper* sente uma obstrução na direção desejada e o que aconteceria era que o somatório das contribuições dos sensores de distância faria com que o *helper* contornasse o obstáculo pelo lado esquerdo o que levava a carga a colidir com o obstáculo. Com esta abordagem fará com que o somatório das contribuições dos sensores de distância entre a carga e a direção que o robô aponta seja sentida no sensor de distância a seguir a esse, isto fará com que o robô se direcione para uma direção mais oportuna.

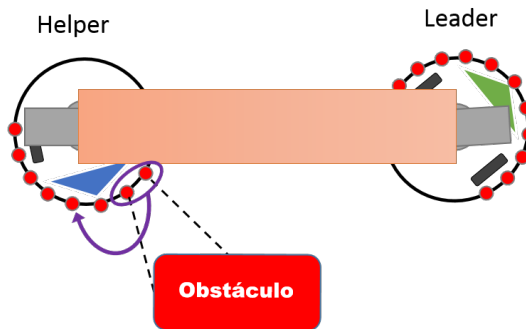


Figura 4.4: Obstáculos detectados entre a direção de navegação do robô *helper* e a carga são movidos para uma posição estratégica, a fim de evitar a colisão da carga com o obstáculo.

No caso do robô *leader*, a componente $f_{obs,L}(\phi_L(t))$, tal como no robô *helper*, permite ao robô evitar os obstáculos ao longo da navegação.

A dinâmica comportamental “evitar obstáculos” resulta do somatório das contribuições de cada sensor de distância na presença de obstáculos (4.13).

$$f_{obs,L}(\phi_L(t)) = \sum_{i=1}^{nSectors} f_{obs,i,L}(\phi_L(t)) \quad (4.13)$$

Nesta componente o $nSectors$ define o número de setores de sensores de distâncias que o robô possui, e $f_{obs,H}(\phi_L(t))$ representa a contribuição do sensor de distância no setor i . Cada contribuição é definida pela equação (4.10), mas considerando a informação do robô *leader*.

A direção onde é detetado o obstáculo $\psi_{obs,i}$ (4.14) por parte de cada sensor (ver Figura 4.5), é definida tendo em consideração a direção de navegação do robô, ϕ_L , e o ângulo que cada sensor é representado, θ_i , estando este numa posição fixa e conhecida em relação à direção de navegação.

$$\psi_{obs,i} = \phi_L + \theta_i \quad (4.14)$$

No caso de $\lambda_{obs,i}$ o seu valor decresce com o aumento da distância, d_i , medida pelo sensor i de acordo com a função:

$$\lambda_{obs,i} = \beta_1 \exp \left[-\frac{d_i}{\beta_2} \right] \quad (4.15)$$

Na equação (4.15) β_1 controla a magnitude máxima da força de repulsão que esta contribuição representa e β_2 controla a sua taxa de decaimento com o aumento da distância.

A gama angular sobre a qual o repulsor exerce sua força repulsiva, σ_i , (4.16) depende do ângulo de sensibilidade do sensor de infravermelhos, $\Delta\theta$, da distância medida, d_i , e do raio do robô, R_{robo} .

$$\sigma_i = \arctan \left[\tan \left(\frac{\Delta\theta}{2} \right) + \frac{R_{robo}}{R_{robo} + d_i} \right] \quad (4.16)$$

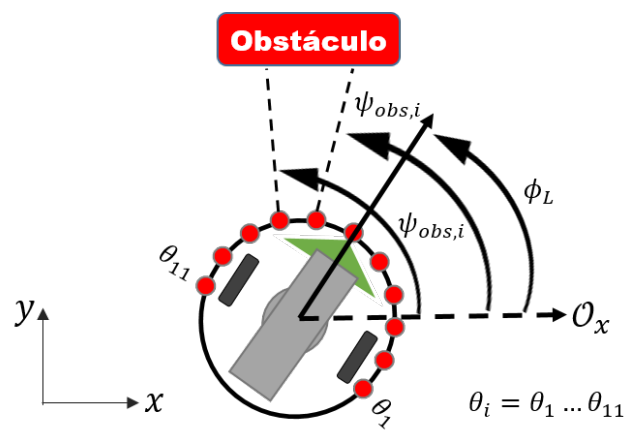


Figura 4.5: Evitar colisões com obstáculos.

Capítulo 5

Estimativa da direção de navegação do robô *leader*

Neste capítulo são apresentados os algoritmos desenvolvidos, e respectivos conceitos técnicos envolvidos, que permitem atingir os objetivos desta dissertação. Na secção 5.1 é apresentado o algoritmo que permite obter uma estimativa da direção de navegação do robô *leader* (ϕ_L) através do cálculo trigonométrico e vetorial. No sentido de melhorar a estimativa, apresentada por este algoritmo, optou-se pela utilização do Filtro de Kalman no modo estimador como é apresentado na secção 5.2.

5.1 Estimativa baseada em cálculo trigonométrico e vetorial

O algoritmo para estimar a direção de navegação do robô *leader*, ϕ_L , presente nesta secção, foi desenvolvido recorrendo ao cálculo trigonométrico e vetorial. A informação sensorial utilizada, nesta abordagem, foi referente ao robô *helper* e ao respetivo suporte. Essa informação foi a posição do robô *helper* no mundo (x_H, y_H), a distância da carga ao centro do robô, d_H , o comprimento da carga a transportar (C_{carga}) e o ângulo da carga em relação ao referencial externo ($\psi_{c,H}$) (ver Figura 5.1).

Inicialmente, considerou-se o caso em que não existe qualquer variação no valor de $\psi_{c,H}$, ou seja, o suporte não sofre qualquer movimento rotacional. Nesta situação verifica-se que a equipa de robôs realiza um movimento segundo uma trajetória retilínea

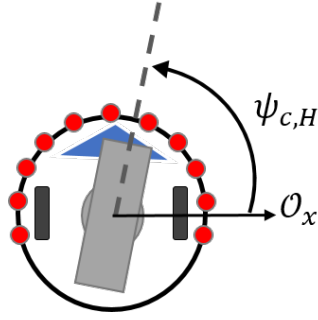


Figura 5.1: Ângulo que a carga faz com o referencial externo.

e a direção de navegação do robô líder mantém-se inalterada. Nesse sentido, considerou-se que quando esta situação é verificada, ao longo da navegação, a estimativa da direção de navegação do robô *leader* ($\phi_L(t)$) mantém-se igual ao valor estimado anteriormente (5.1).

$$\phi_{L_k} = \phi_{L_{k-1}} \quad (5.1)$$

Quando se verificam alterações no valor de $\psi_{c,H}$, resultando numa mudança na direção de navegação do robô *leader*, a estimativa da direção atual, é obtida a partir do cálculo trigonométrico e vetorial, como foi mencionado no início do capítulo. Começou-se por calcular a diferença existente, entre a posição do robô *leader* e a do robô *helper*, em relação ao eixo horizontal (5.2) e vertical (5.3) (ver Figura 5.2).

$$\Delta x = \cos(\psi_{c,H}) \cdot d_{robos} \quad (5.2)$$

$$\Delta y = \sin(\psi_{c,H}) \cdot d_{robos} \quad (5.3)$$

A distância entre os robôs, d_{robos} , (5.4) é obtida tendo em conta o comprimento da carga (C_{carga}) e a distância da carga em relação ao centro do robô *leader* (d_L) e do *helper* (d_H) (ver Figura 5.2). Como não se tem acesso à informação sensorial do robô *leader* considerou-se no cálculo de d_{robos} que a distância da carga em relação ao centro do robô *leader* é igual à do robô *helper*.

$$d_{robos} = C_{carga} + 2d_H \quad (5.4)$$

A partir do valor da diferença entre a posição, do robô *leader* e a do robô *helper*, em relação aos dois eixos (horizontal e vertical) é possível obter uma estimativa da posição

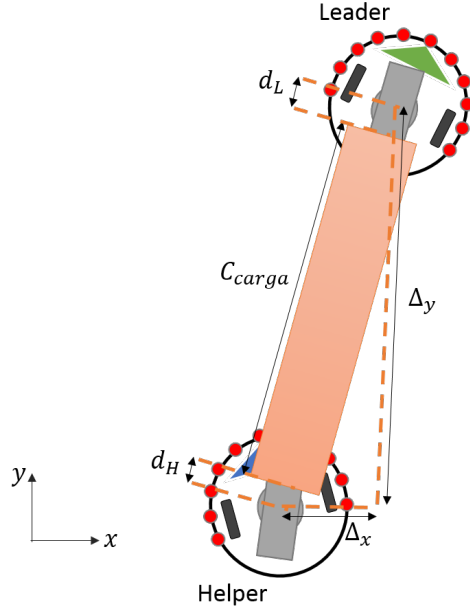


Figura 5.2: Diferença entre a posição do robô *leader* e a posição do robô *helper* em ambos os eixos (horizontal e vertical)

do *leader* no mundo (x_L, y_L) , tanto no instante anterior (5.5) (5.6) como no instante atual (5.7) (5.8).

$$x_{L_{k-1}} = x_{H_{k-1}} + \Delta x_{k-1} \quad (5.5)$$

$$y_{L_{k-1}} = y_{H_{k-1}} + \Delta y_{k-1} \quad (5.6)$$

$$x_L = x_{H_k} + \Delta x_k \quad (5.7)$$

$$y_L = y_{H_k} + \Delta y_k \quad (5.8)$$

A partir da estimativa da posição do robô *leader*, é possível obter uma estimativa, da sua direção de navegação, através do cálculo vetorial. A orientação do vetor, definido pela componente horizontal (5.9) e pela componente vertical (5.10), representa o valor estimado para a direção de navegação do robô *leader* (5.11). A componente horizontal é o resultado da diferença entre posição anterior e atual do robô *leader* no eixo horizontal e a componente vertical é o resultado da diferença entre posição anterior e atual do robô *leader* no eixo vertical.

$$dx = x_{L_k} - x_{L_{k-1}} \quad (5.9)$$

$$dy = y_{L_k} - y_{L_{k-1}} \quad (5.10)$$

$$\phi_{L_k} = \text{atan2}(d_y, d_x) \quad (5.11)$$

Porém esta abordagem nem sempre apresentou uma estimativa da direção de navegação do robô *leader* o mais correta possível. Isto acontecia quando se verificava variações bruscas na estimativa da velocidade angular. A solução encontrada foi a utilização dos filtros de Kalman como é descrito na secção 5.2.

5.2 Filtro de Kalman Estendido

No sentido de melhorar a estimativa apresentada pelo algoritmo, presente na secção 5.1, para a estimativa da direção de navegação do robô *leader* (ϕ_L), optou-se pela utilização do Filtro de Kalman Estendido. A escolha da utilização do Filtro de Kalman Estendido, deve-se ao facto de ser uma das técnicas muito utilizada na estimativa de um estado, a partir de medições que contém ruídos e incertezas ao longo do tempo.

5.2.1 Aplicação do Filtro de Kalman Estendido na estimativa da direção de navegação do robô *leader*

Na aplicação do Filtro de Kalman Estendido o modelo definido resultou num sistema dinâmico não linear de cinco estados. Os estados que serão previstos na etapa de previsão e posteriormente corrigidos na etapa de correção do filtro são: a posição em relação ao eixo horizontal, x , a posição em relação ao eixo vertical, y , a velocidade linear, v , a orientação (medida em relação ao referencial externo), ϕ , e a velocidade angular do robô *leader*, Ω , resultando na matriz de estado representada em (5.12).

$$X = \begin{bmatrix} x \\ y \\ v \\ \phi \\ \Omega \end{bmatrix} \quad (5.12)$$

A velocidade linear do robô (5.13) considerou-se que será constante ao longo do tempo.

$$v_k = v_{k-1} \quad (5.13)$$

A posição do robô *leader* no instante atual em relação ao eixo horizontal (5.14) e vertical (5.15) resultará da integração da velocidade linear do robô em ordem ao tempo.

$$x_k = x_{k-1} + v_{k-1} \Delta t \cos(\phi_{k-1}) \quad (5.14)$$

$$y_k = y_{k-1} + v_{k-1} \Delta t \sin(\phi_{k-1}) \quad (5.15)$$

A velocidade angular (5.16) considerar-se-á constante ao longo do tempo.

$$\Omega_k = \Omega_{k-1} \quad (5.16)$$

A orientação (5.17) resultará da integração da velocidade angular em ordem ao tempo.

$$\phi_k = \phi_{k-1} + \Omega_{k-1} \Delta t \quad (5.17)$$

5.2.1.1 Etapa de previsão

Na etapa de previsão do filtro o estado é previsto tendo em consideração a matriz de transição, Φ_k , que neste caso é uma expressão não linear do modelo, resultando na matriz representada em (5.18).

$$\Phi_k = \begin{bmatrix} 1 & 0 & \Delta t \cos(\phi_{k-1}) & 0 & 0 \\ 0 & 1 & \Delta t \sin(\phi_{k-1}) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

No caso da previsão da estimativa do erro de covariância esta é modificada pela forma linearizada do modelo, através do Jacobiano, que neste caso resulta na matriz

representada em (5.19).

$$F_k = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \Phi} & \frac{\partial f_1}{\partial \Omega} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \Phi} & \frac{\partial f_2}{\partial \Omega} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \Phi} & \frac{\partial f_3}{\partial \Omega} \\ \frac{\partial f_4}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial v} & \frac{\partial f_4}{\partial \Phi} & \frac{\partial f_4}{\partial \Omega} \\ \frac{\partial f_5}{\partial x} & \frac{\partial f_5}{\partial y} & \frac{\partial f_5}{\partial v} & \frac{\partial f_5}{\partial \Phi} & \frac{\partial f_5}{\partial \Omega} \\ x & y & v & \Phi & \Omega \end{bmatrix} \Longleftrightarrow \begin{bmatrix} 1 & 0 & \Delta t \cos(\phi_{k-1}) & -v_{k-1} \Delta t \sin(\phi_{k-1}) & 0 \\ 0 & 1 & \Delta t \sin(\phi_{k-1}) & -v_{k-1} \Delta t \cos(\phi_{k-1}) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.19)$$

A matriz de covariância do estado, Q_k , é uma matriz diagonal em que os elementos da diagonal são os erros associados a cada estado (5.20).

$$Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\Omega^2 \end{bmatrix} \quad (5.20)$$

O erro associado ao modelo é em grande parte devido ao facto de se ter considerado que as velocidades (linear e angular) são constantes ao longo do tempo. Nesse sentido, o erro associado ao estado da velocidade (5.21) foi definido em função da aceleração máxima, pois corresponde ao maior erro possível:

$$\sigma_v = \Delta t a_{max} \quad (5.21)$$

Neste caso, definiu-se que o valor da aceleração máxima era $1m/s^2$. No caso do erro associado ao estado da posição, x e y , este obtêm-se pela integração do erro associado ao estado da velocidade (5.22) (5.23).

$$\sigma_x = \frac{\Delta t^2 a_{max}}{2} \quad (5.22)$$

$$\sigma_y = \frac{\Delta t^2 a_{max}}{2} \quad (5.23)$$

O erro associado à velocidade angular (5.24) é definido em função da aceleração angular máxima, que neste caso, definiu-se que o valor era $1rad/s^2$. O erro associado ao estado da orientação (5.25) é obtido pela integração do erro associado ao estado da velocidade angular, como é apresentado nas seguintes equações.

$$\sigma_\Omega = \Delta t \alpha_{max} \quad (5.24)$$

$$\sigma_\Phi = \frac{\Delta t^2 \alpha_{max}}{2} \quad (5.25)$$

5.2.1.2 Etapa de correção

Na etapa de correção do estado, o ganho de Kalman é utilizado para corrigir a estimativa do estado e reduzir a covariância deste. O valor, do ganho de Kalman, é obtido a partir da estimativa do erro, P_k^- , da matriz de medição, H_k , e da matriz de covariância da medição do sensor, R_k .

A matriz de covariância da medição do sensor, R_k , será uma matriz diagonal, onde os elementos da sua diagonal, são os erros associados a cada medição do sensor (5.26).

$$R_k = \begin{bmatrix} R_x & 0 & 0 & 0 & 0 \\ 0 & R_y & 0 & 0 & 0 \\ 0 & 0 & R_v & 0 & 0 \\ 0 & 0 & 0 & R_\phi & 0 \\ 0 & 0 & 0 & 0 & R_\Omega \end{bmatrix} \quad (5.26)$$

Na correção do estado são utilizadas as medições obtidas, para os cinco estados, pelo algoritmo apresentado na secção 5.1.

5.3 Implementação do módulo para obter o valor do ângulo que a carga faz com a direção de navegação do robô *leader*

O módulo (ver Figura 5.3) que permite ao robô *helper* obter uma medida do ângulo que a carga faz com a direção de navegação do robô *leader* foi implementado em Matlab. O módulo permite que a aplicação de controlo do robô *helper* envie e receba dados através de comandos pré-definidos, utilizando uma tipologia cliente - servidor. Neste caso a aplicação de controlo do robô é o cliente e o módulo implementado é o servidor. A informação enviada da aplicação de controlo do robô *helper* para o módulo é: a posição do robô no mundo (x_H, y_H), a distância da carga ao centro do robô (d_H), o comprimento da carga (C_{carga}), o ângulo da carga em relação ao referencial externo ($\psi_{c,H}$) e o intervalo de tempo decorrido entre iterações (Δt). Como resposta o módulo envia uma medida do ângulo que a carga faz com a direção de navegação do robô *leader*.

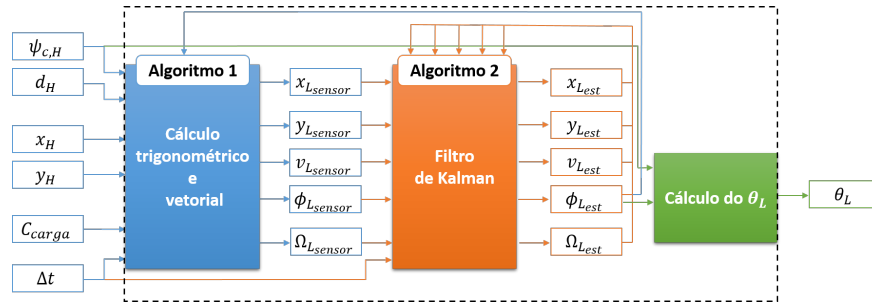


Figura 5.3: Arquitetura do módulo implementado.

5.3.1 Algoritmo baseado em cálculo trigonométrico e vetorial

O algoritmo 1 funcionará como sensor no módulo implementado, fornecendo uma estimativa da posição ($x_{Lsensor}, y_{Lsensor}$), da orientação ($\phi_{Lsensor}$) e da velocidade angular e linear ($v_{Lsensor}, \Omega_{Lsensor}$) do robô *leader*. O valor do intervalo de tempo, Δt , é calculado a cada ciclo de controlo do processo. Este valor, do intervalo de tempo, será o real para que a informação seja mais confiável.

Algoritmo 1: Medida da direção de navegação do robô *leader*

Data: $x_H, y_H, d_H, \psi_{c,H}, C_{carga}, \Delta t$

Result: $x_{L_{sensor}}, y_{L_{sensor}}, v_{L_{sensor}}, \phi_{L_{sensor}}, \Omega_{L_{sensor}}$

begin

$$d_{robos_{k-1}} = C_{carga} + 2d_{H_{k-1}}$$

$$d_{robos_k} = C_{carga} + 2d_{H_k}$$

// Cálculo da diferença entre a posição do robô leader e a do robô helper

$$\Delta x_{k-1} = \cos(\psi_{c,H}).d_{robos_{k-1}}$$

$$\Delta y_{k-1} = \sin(\psi_{c,H}).d_{robos_{k-1}}$$

$$\Delta x_k = \cos(\psi_{c,H}).d_{robos_k}$$

$$\Delta y_k = \sin(\psi_{c,H}).d_{robos_k}$$

// Cálculo da posição do robô leader no mundo (x_L, y_L)

$$x_{L_{k-1}} = x_{H_{k-1}} + \Delta x_{k-1}$$

$$y_{L_{k-1}} = y_{H_{k-1}} + \Delta y_{k-1}$$

$$x_{L_k} = x_{H_k} + \Delta x_k$$

$$y_{L_k} = y_{H_k} + \Delta y_k$$

$$dx = x_{L_k} - x_{L_{k-1}}$$

$$dy = y_{L_k} - y_{L_{k-1}}$$

// Cálculo da direção de navegação ϕ_L

if $\psi_{c,H_k} == \psi_{c,H_{k-1}}$ **then**

$$\quad \phi_{L_k} = \phi_{L_{k-1}}$$

else

$$\quad \phi_{L_k} = \arctan(dy, dx)$$

// Cálculo da velocidade linear e angular

$$v_x = \frac{dx}{\Delta t}$$

$$v_y = \frac{dy}{\Delta t}$$

$$v_L = \sqrt{v_x^2 + v_y^2}$$

$$\Omega_L = \frac{\phi_{L_k} - \phi_{L_{k-1}}}{\Delta t}$$

5.3.2 Algoritmo da implementação do Filtro de Kalman Estendido

No algoritmo 2 está implementado o filtro de Kalman Estendido no modo estimador.

Na implementação do filtro deve-se ter especial atenção na sintonização dos valores dos pesos, da matriz de covariância do estado, Q_k , e da matriz de covariância da medição do sensor, R_k , devido, ao facto destes parâmetros, terem uma grande influência na performance do filtro.

Algoritmo 2: Estimativa da direcção de navegação do robô *leader* utilizando o Filtro de Kalman Estendido

Data: $x_{L_{sensor}}, y_{L_{sensor}}, v_{L_{sensor}}, \phi_{L_{sensor}}, \Omega_{L_{sensor}}, \Delta t$

Result: $x_{L_{est}}, y_{L_{est}}, v_{L_{est}}, \phi_{L_{est}}, \Omega_{L_{est}}$

begin

 // Inicialização das matrizes do filtro

$F, X_0, P_0, \Phi, H, R, Q$

 // Etapa 1: Previsão

 // Previsão do estado

$X = \Phi * X_0$

 // Previsão da covariância

$P = F * P_0 * F' + Q$

 // Etapa 2: Correção

$In = [x_{L_{sensor}} - X[1]; y_{L_{sensor}} - X[2]; v_{L_{sensor}} - X[3]; \phi_{L_{sensor}} - X[4]; \Omega_{L_{sensor}} - X[5]]$

 // Ganho de Kalman

$K = (P * H') / (H * P * H' + R)$

 // Atualização do estado

$X = X + k * In$

 // Atualização da covariância de estado

$P = (I - K * H) * P$

 // Parâmetros estimados

$x_{L_{est}} = X[1]; y_{L_{est}} = X[2]; v_{L_{est}} = X[3]; \phi_{L_{est}} = X[4]; \Omega_{L_{est}} = X[5]$

5.3.3 Cálculo do θ_L

Neste momento a informação que é efetivamente comunicada entre a equipa de robôs, não é a direção de navegação do robô *leader* (ϕ_L), mas o ângulo que a carga faz com a atual direção de navegação do robô *leader* (θ_L). Por esse motivo, foi necessário obter o valor desse parâmetro (5.27), tendo em conta o valor estimado para a direção de navegação, resultante do filtro de Kalman Estendido.

$$\theta_L = \psi_{c,L} - \phi_{L_{est}} \quad (5.27)$$

Onde o parâmetro $\psi_{c,L}$ (ver Figura 5.4) é obtido a partir do parâmetro $\psi_{c,H}$ referente ao robô *helper*, é dado por:

$$\psi_{c,L} = \psi_{c,H} + \pi$$

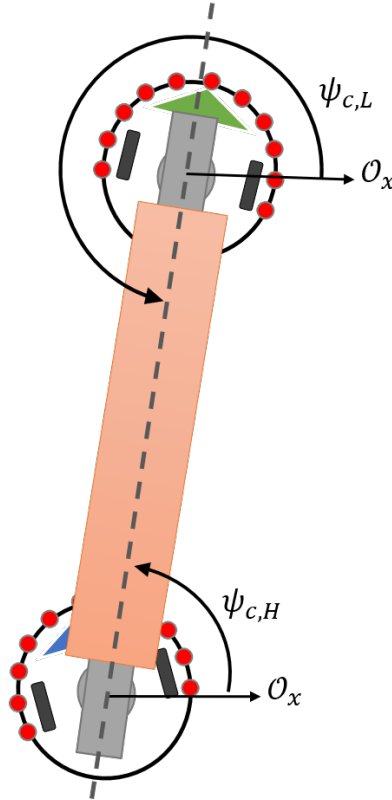


Figura 5.4: Ângulo que a carga faz com o referencial externo do ponto de vista de ambos os robôs.

Esta página, intencionalmente, apenas contém esta frase.

Capítulo 6

Resultados

Neste capítulo são apresentados os principais resultados obtidos na realização dos testes experimentais. Primeiramente serão apresentados os resultados obtidos no ambiente de simulação. Em seguida serão apresentados os resultados no ambiente real do algoritmo implementado, que foram realizados nas instalações da Universidade do Minho. E por fim será feita uma análise aos resultados obtidos nos testes realizados.

6.1 Apresentação dos resultados obtidos

Nos testes realizados pretendia-se avaliar a eficiência do algoritmo no cálculo do ângulo que a carga faz com a direção atual do robô *leader*, θ_L , a partir da estimativa da direção de navegação do robô *leader* por parte do robô *helper*, sem que a tarefa de transporte fosse comprometida. Nesse sentido, escolheram-se ambientes que permitiam simular diferentes situações.

Para cada teste realizado é apresentado um gráfico, que contem a seguinte informação:

- Pontos de cor magenta: representam o valor do ângulo que a carga faz com a direção atual do robô *leader*, obtido pelo algoritmo implementado;
- Linha de cor verde: representa o valor real do ângulo que a carga faz com a direção atual do robô *leader*, para fim de comparação, obtido pela leitura do *encoder* presente no suporte da carga.

Adicionalmente também é apresentada para cada simulação a trajetória realizada pela equipa até atingir a posição de destino.

Na apresentação dos resultados apenas é exposta a informação do valor do ângulo que a carga faz com a direção atual do robô *leader*, a partir da estimativa da direção de navegação do robô. Não é apresentada toda a informação que é estimada (posição e velocidade), devido ao facto de apenas esta ser considerada na arquitetura de controlo do robô *helper*, durante a tarefa de transporte.

6.1.1 Testes realizados no ambiente de simulação CoopDynSim

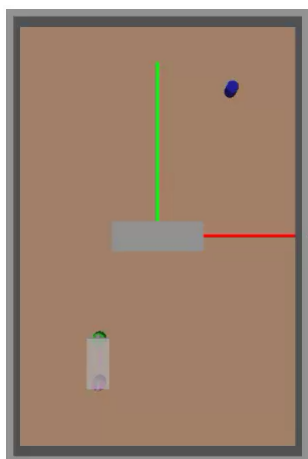
Nos testes realizados no ambiente de simulação é analisada a performance do algoritmo implementado em oito cenários distintos.

Nos oito cenários considerou-se que as dimensões da carga a transportar pela equipa de robôs eram: 1.5 m de comprimento e 0.75 m de largura. E as condições iniciais definidas para o robô *leader* foram: $x_{L_0} = 0$ m, $y_{L_0} = 0$ m, $v_{L_0} = 0$ m/s, $\phi_{L_0} = 90^\circ$, $\Omega_{L_0} = 0$ rad/s.

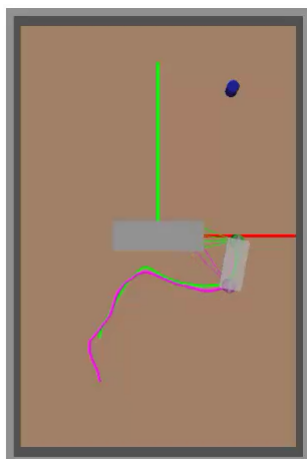
6.1.1.1 Cenário 1

O cenário 1 (Figura 6.1a) corresponde a um ambiente de uma sala retangular com um obstáculo no centro. A equipa de robôs tem de contornar o obstáculo para chegar ao alvo presente na sala (alvo de cor azul). Neste cenário, a equipa de robôs parte segundo uma orientação de 90 graus, em direção ao alvo que se encontra na posição de coordenadas $(x, y, z) = (2.5, 5, 0)$.

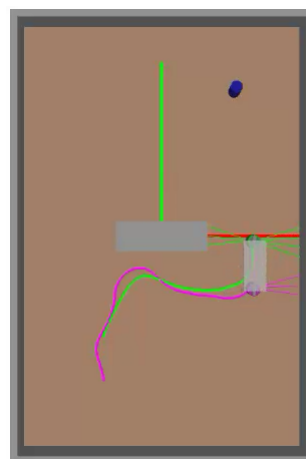
Para este cenário realizaram-se duas simulações, uma em que existia comunicação entre o *leader* e o *helper*, do valor do ângulo que a carga faz com a direção atual do robô *leader*, e outra em que essa comunicação foi eliminada. Na Figura 6.1 observa-se o movimento realizado pela equipa de robôs desde a posição inicial até atingir o alvo para ambas as situações. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.1a), quando a equipa contorna o obstáculo (Figura 6.1b) (Figura 6.1c), presente na sala, e ainda quando atinge o alvo (Figura 6.1d) (Figura 6.1e).



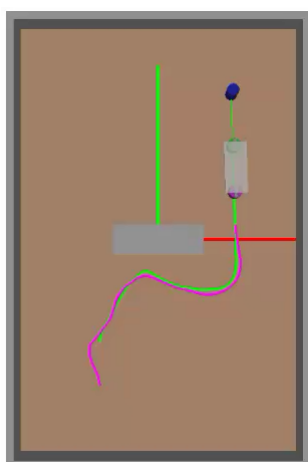
(a) Equipa inicia o movimento ($t = 0$ s).



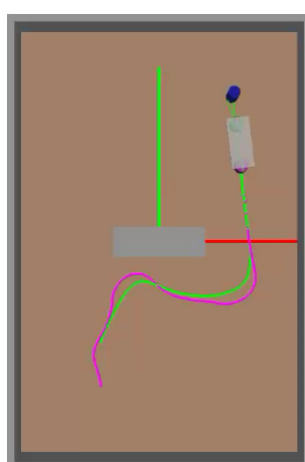
(b) Equipa de robôs contorna obstáculo (com comunicação).



(c) Equipa de robôs contorna obstáculo (sem comunicação).



(d) Equipa de robôs atinge o alvo (com comunicação).



(e) Equipa de robôs atinge o alvo (sem comunicação).

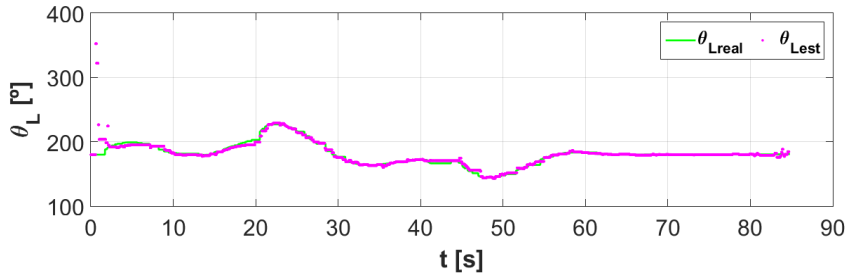
Figura 6.1: Movimento da equipa até atingir o alvo.

Na análise às trajetórias realizadas pela equipa em ambas as situações (Figura 6.1) verifica-se que a diferença entre uma situação e a outra é mínima.

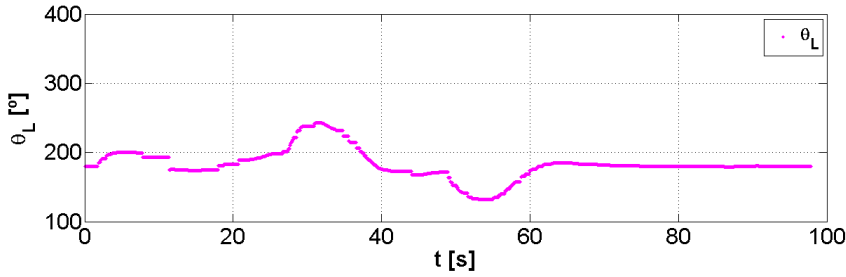
Na Figura 6.2, observa-se a evolução do ângulo que a carga faz com a direção atual do robô *leader*, θ_L , ao longo do tempo para ambas as situações. A análise aos resultados obtidos para a situação que não existe comunicação (Figura 6.2a) permite aferir que o valor calculado acompanha de perto o valor real, verificando-se que a tarefa de transporte não é comprometida. De salientar apenas que é necessário um certo tempo ($\approx 3s$) para que o algoritmo estabilize, mas que não afeta o comportamento da equipa na tarefa de transporte.

Observando o gráfico (Figura 6.2a) verifica-se que em $t = 0$ s o ângulo que a carga faz com a direção atual do robô *leader* é de aproximadamente 180 graus e que o mesmo vai variando atingindo o ponto máximo de 227 graus em $t = 22$ s, um ponto mínimo de 145 graus em $t = 48$ s e que a partir de $t = 61$ s o valor do ângulo estabiliza nos 180 graus até chegar ao alvo.

No gráfico obtido para a situação que existe comunicação (Figura 6.2b) observa-se uma ligeira diferença nos valores do ângulo em relação a que não existe comunicação, nomeadamente, no valor máximo que neste caso alcança um valor de 243 graus e no valor mínimo que atinge 132 graus.



(a) Ângulo que a carga faz com a direção atual do robô *leader* (sem comunicação).



(b) Ângulo que a carga faz com a direção atual do robô *leader* (com comunicação).

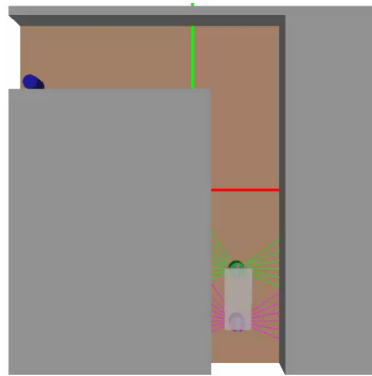
Figura 6.2: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 1.

6.1.1.2 Cenário 2

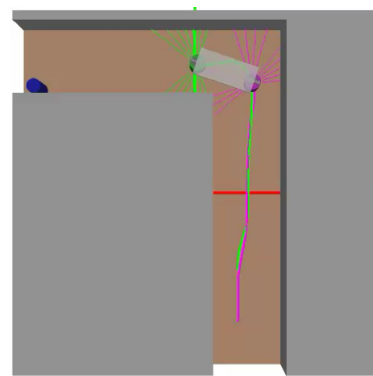
No cenário 2 (Figura 6.3a), a equipa tem que navegar num corredor com um canto de 90 graus, que terá de contornar para chegar ao alvo (cor azul). Neste cenário, a equipa de robôs parte segundo uma orientação de 90 graus, para chegar ao alvo, que se encontra na posição de coordenadas $(x, y, z) = (-4, 3, 0)$.

Tal como no cenário anterior, também para este cenário se realizaram duas simulações, uma em que existe comunicação entre a equipa e na outra em que essa comunicação foi eliminada.

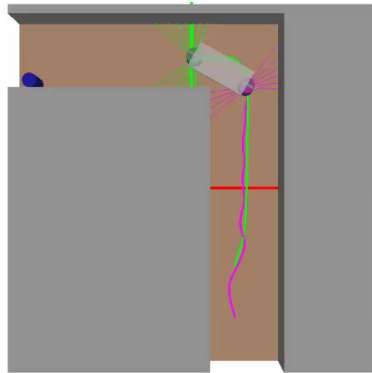
Na Figura 6.3 observa-se o movimento realizado pela equipa de robôs para ambas as situações. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o seu movimento (Figura 6.3a), quando a equipa contorna o canto (Figura 6.3b) (Figura 6.3c) e ainda quando atinge o alvo (Figura 6.3d) (Figura 6.3e). Na análise às trajetórias realizadas pela equipa em ambas as situações (Figura 6.3), verifica-se que a diferença existente entre uma situação e a outra é mínima.



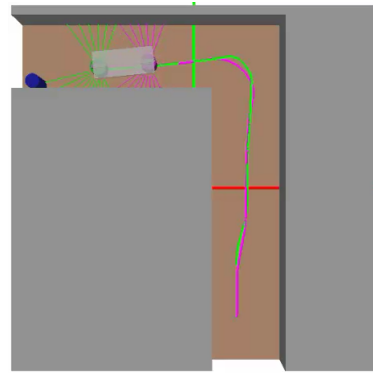
(a) Equipa inicia o movimento ($t = 0$ s).



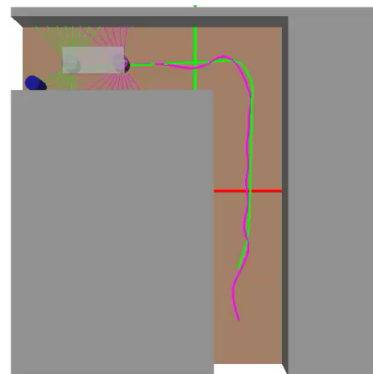
(b) Equipa de robôs contorna o canto (com comunicação).



(c) Equipa de robôs contorna o canto (sem comunicação).



(d) Equipa de robôs atinge o alvo (com comunicação).



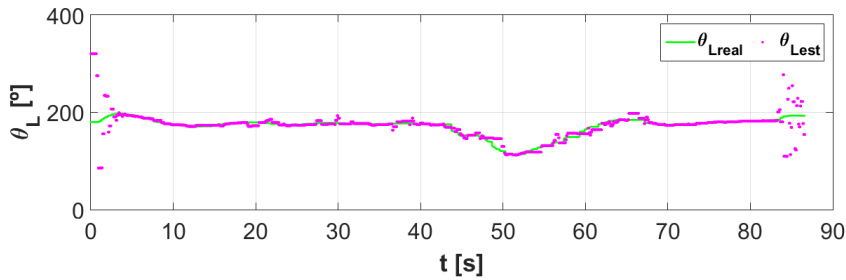
(e) Equipa de robôs atinge o alvo (sem comunicação).

Figura 6.3: Movimento da equipa até atingir o alvo.

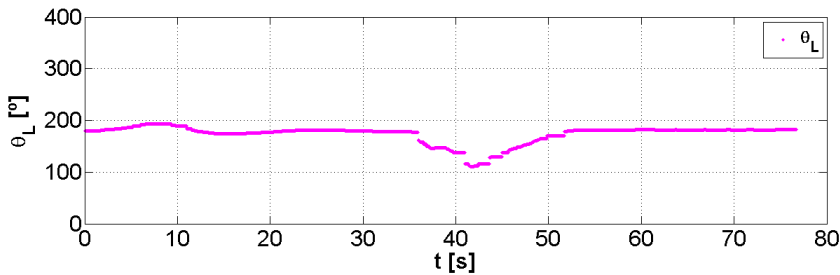
Na Figura 6.4, observa-se a evolução do ângulo que a carga faz com a direção atual do robô *leader*, θ_L , ao longo do tempo para ambas as situações.

Na análise aos resultados obtidos para a situação que não existe comunicação (ver Figura 6.4a), tal como no cenário anterior, a tarefa de transporte não é comprometida o que viabiliza o algoritmo implementado.

No gráfico obtido para a situação que existe comunicação (Figura 6.4b) observa-se uma pequena diferença nos valores do ângulo em relação à outra situação, nomeadamente, no valor máximo que neste caso é inferior (193 graus) à que não existe comunicação (197 graus) e no valor mínimo atinge um valor mais baixo (110 graus) do que na outra situação (114 graus).



(a) Ângulo que a carga faz com a direção atual do robô *leader* (sem comunicação).



(b) Ângulo que a carga faz com a direção atual do robô *leader* (com comunicação).

Figura 6.4: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 2.

6.1.1.3 Cenário 3

No cenário 3 (Figura 6.5a), a equipa de robôs tem que navegar sobre um corredor em forma de U invertido. Para chegar ao alvo (cor azul), a equipa terá de contornar o obstáculo e depois navegar em sentido oposto. Neste cenário, a equipa de robôs, parte

segundo a orientação de 90 graus, contorna o obstáculo, para chegar ao alvo, que se encontra na posição de coordenadas $(x, y, z) = (-0.75, -1, 0)$.

O movimento realizado pela equipa de robôs desde a posição inicial até atingir o alvo é observado na Figura 6.5. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.5a), quando contorna o obstáculo e começa a navegar em sentido oposto (Figura 6.5b) e ainda quando atinge o alvo (Figura 6.5c).

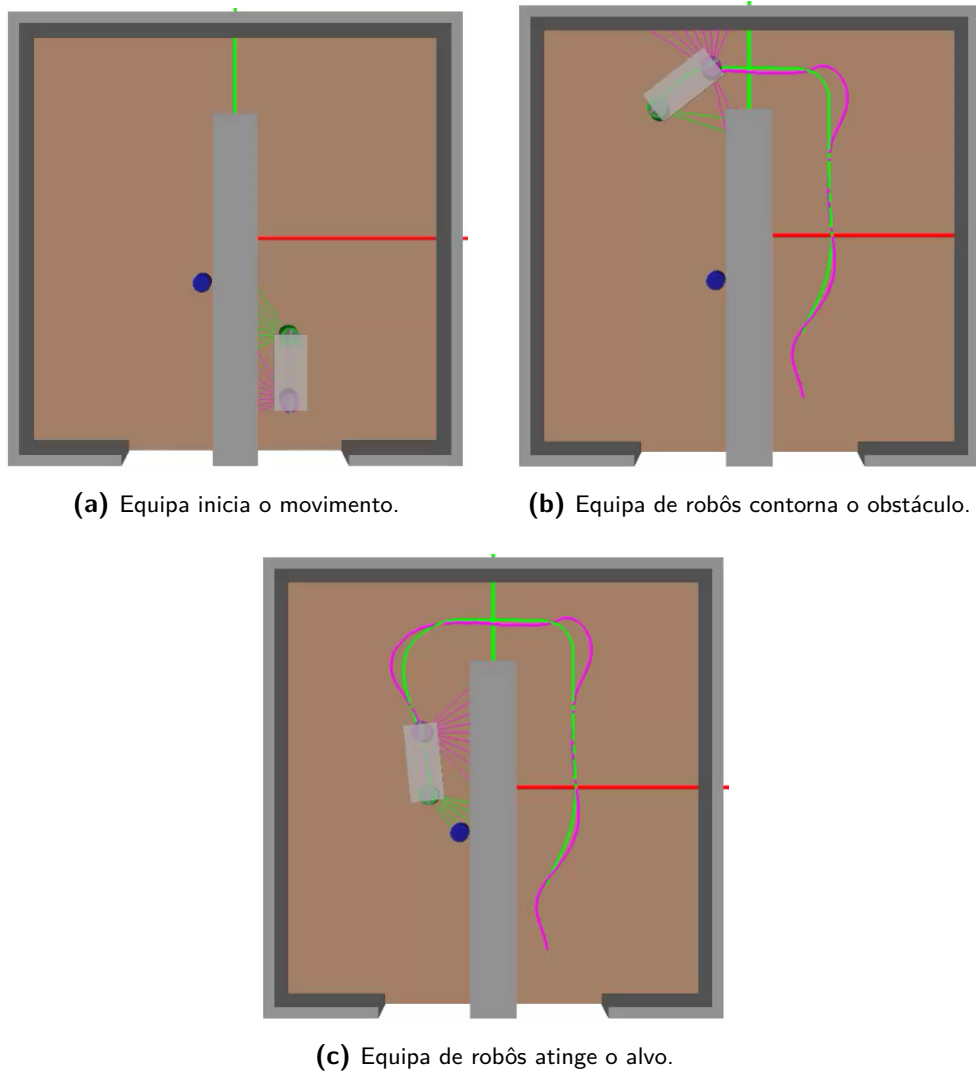


Figura 6.5: Movimento da equipa até atingir o alvo.

Para este cenário, os resultados obtidos (ver Figura 6.6) demonstram mais uma vez a viabilidade do algoritmo implementado no comportamento do robô *helper* durante a tarefa de transporte.

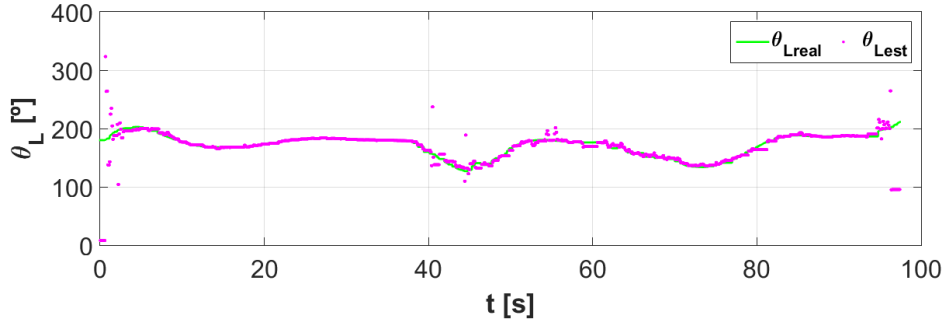


Figura 6.6: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 3.

Adicionalmente, realizou-se para este cenário uma simulação que se estudou a influência do valor atribuído à distância da carga em relação ao centro do robô *leader* no cálculo da distância entre os robôs. Como foi apresentado na secção 5.1, na estimativa da direção de navegação do robô *leader* é considerada a distância entre os robôs, sendo esta obtida tendo em conta o comprimento da carga e a distância da carga em relação ao centro do robô *leader* e do *helper*. Porém, o valor da distância da carga em relação ao centro do robô *leader* não está acessível por isso considerou-se que a distância da carga em relação ao centro do robô *leader* era igual à do robô *helper*. No sentido de averiguar a influência desse valor no cálculo de d_{robos} , realizou-se para este cenário uma simulação que se considerou uma variação desse valor até 200 mm, independente do valor de d_H . Pela análise aos resultados obtidos (ver Figura 6.7) verifica-se que esta variação no valor de d_H provocou um ligeiro atraso na estimativa mas que não prejudicou o comportamento do robô *helper* na realização da tarefa.

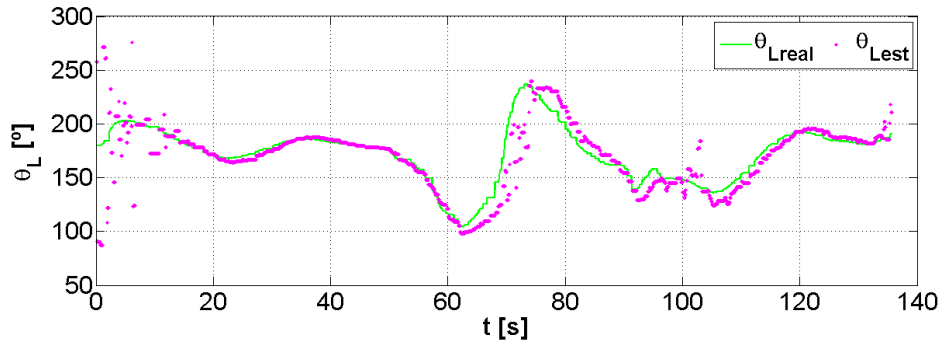


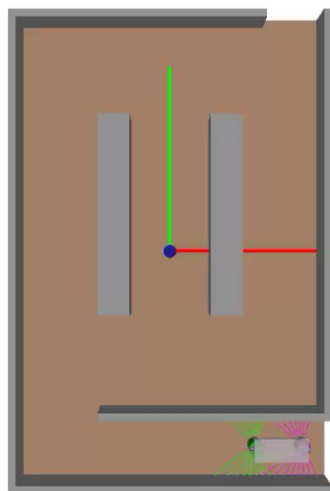
Figura 6.7: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 3.

6.1.1.4 Cenário 4

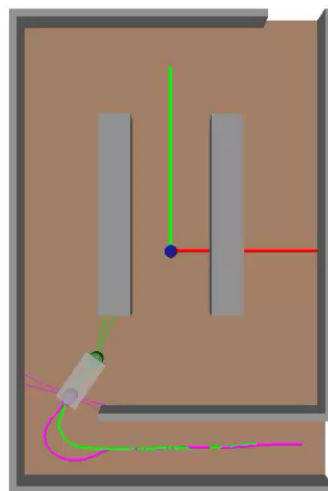
No cenário 4 (Figura 6.8a), a equipa de robôs, tem uma dificuldade superior, pois parte de um corredor separado por uma parede, tendo que a contornar para chegar ao alvo, que está colocado no centro entre dois muros paralelos. Neste cenário, a equipa de robôs parte segundo uma orientação de 180 graus, ao encontro do alvo que se encontra na posição de coordenadas $(x, y, z) = (0, 0, 0)$.

Na Figura 6.8 observa-se o movimento realizado pela equipa de robôs desde a posição inicial até atingir o alvo. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.8a), quando atravessa o corredor e se dirige para o alvo (Figura 6.8b) e ainda quando atinge a posição do alvo (Figura 6.8c).

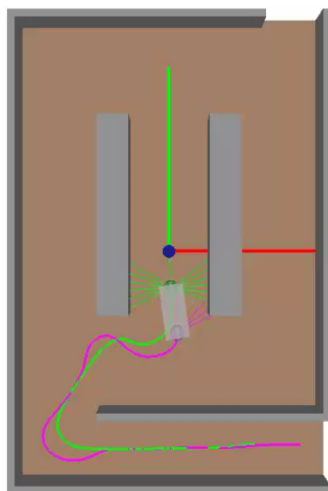
Os resultados obtidos (Figura 6.9) para este cenário continuam a demonstrar a fiabilidade do algoritmo implementado no cálculo da medida do ângulo que a carga faz com a direção atual do robô *leader*.



(a) Equipa inicia o movimento
($t = 0$ s).



(b) Equipa de robôs atravessa o
corredor.



(c) Equipa de robôs atinge o alvo.

Figura 6.8: Movimento da equipa até atingir o alvo.

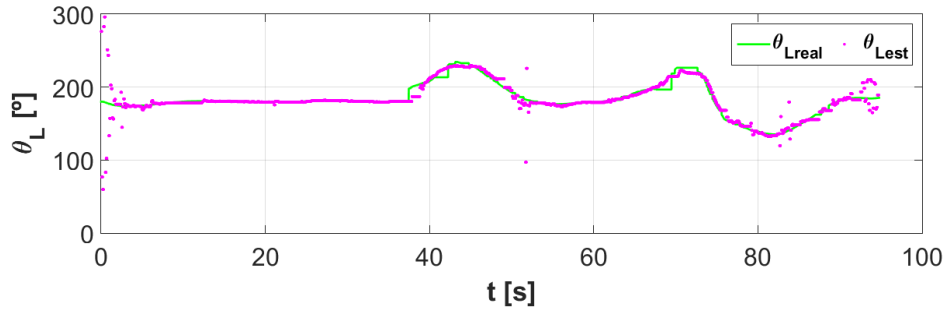


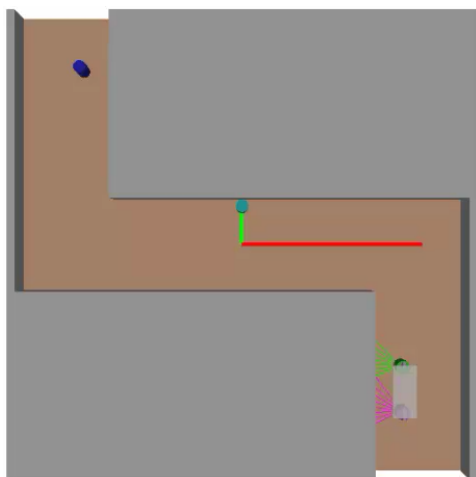
Figura 6.9: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 4.

6.1.1.5 Cenário 5

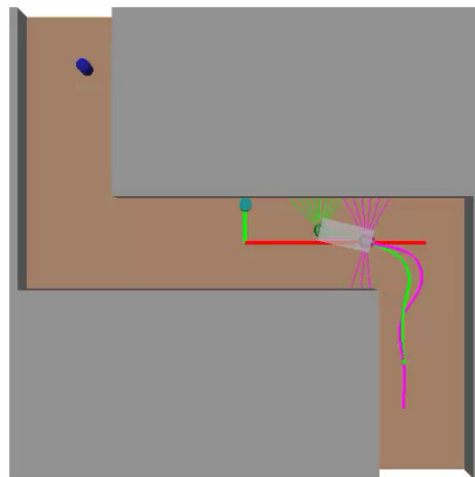
No cenário 5 (Figura 6.10a), a equipa de robôs, tem que percorrer um corredor com dois cantos de 90 graus. Para este cenário existe a particularidade de não haver apenas um alvo, mas dois alvos que a equipa terá de atingir. Neste cenário, a equipa de robôs parte segundo uma orientação de 90 graus, ao encontro de um primeiro alvo (cor ciano) que se encontra na posição de coordenadas $(x, y, z) = (0, 1.25, 0)$ e de um segundo alvo (cor azul) que se encontra na posição de coordenadas $(x, y, z) = (-5.25, 5.75, 0)$.

O movimento realizado pela equipa de robôs desde da posição inicial até atingir o último alvo pode ser observável na Figura 6.10. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.10a), quando contorna o primeiro canto e se dirige para o primeiro alvo (Figura 6.10b), presente na sala, o momento em que o robô *leader* pára (Figura 6.10c) e ainda quando a equipa atinge o segundo alvo (Figura 6.10d).

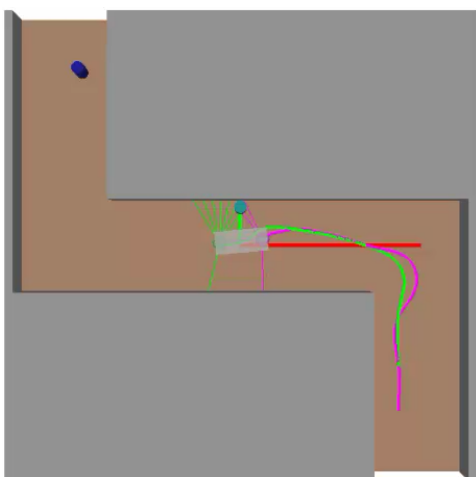
Para este cenário, em particular, foi estudada a performance do algoritmo implementado no caso em que o robô *leader* pára, resultando numa velocidade nula. Nos resultados obtidos (ver Figura 6.11) é evidenciado que o algoritmo não foi capaz de dar uma estimativa correta no intervalo de tempo (entre $t = 66$ s e $t = 79$ s) que o robô *leader* esteve parado, mas durante o tempo que a equipa se encontrava em movimento o algoritmo revelou-se eficiente no cálculo do ângulo que a carga faz com a direção atual do robô *leader*. O problema evidenciado nesta situação prevê-se que é devido ao facto, de durante o intervalo de tempo que a equipa não se encontra em movimento



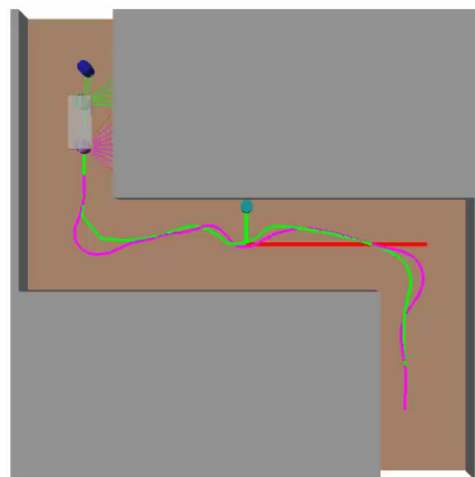
(a) Equipa inicia o movimento ($t = 0$ s).



(b) Equipa de robôs contorna o canto e dirige-se para o primeiro alvo.



(c) Robô *leader* pára.



(d) Equipa de robôs atinge o segundo alvo.

Figura 6.10: Movimento da equipa até atingir o último alvo.

haja ocorrência de variações no ângulo da carga em relação ao referencial externo. Essas variações implicam a não estabilização do valor do ângulo, visto que este parâmetro no algoritmo implementado tem influência direta na estabilização desse valor, para situações em que a direção de navegação do robô *leader* se mantém inalterada, como é o caso nesta situação.

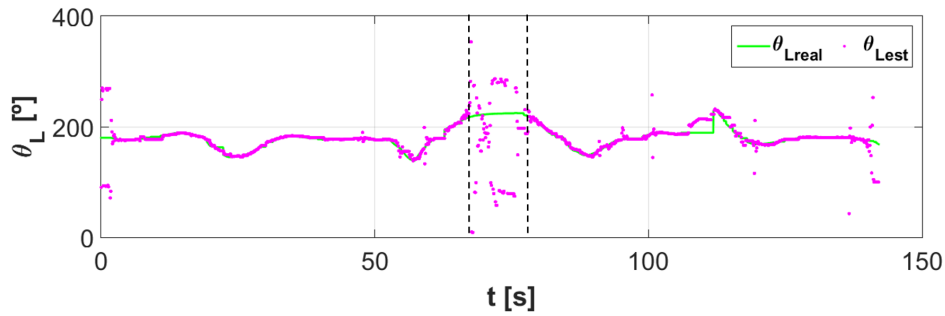
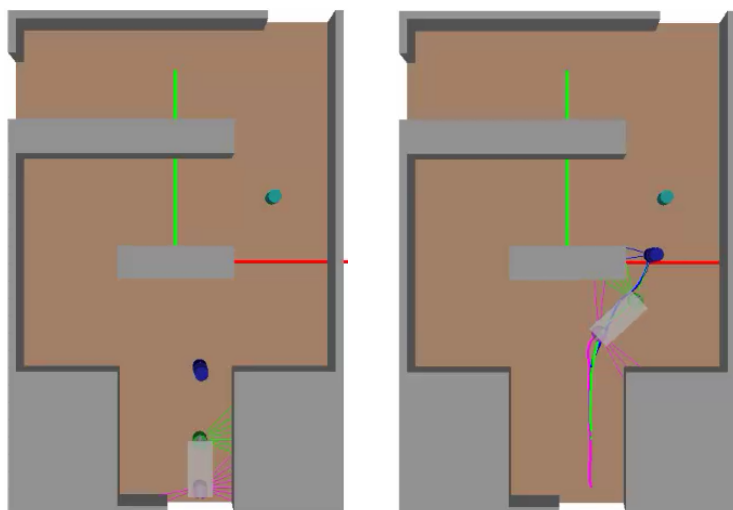


Figura 6.11: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 5.

6.1.1.6 Cenário 6

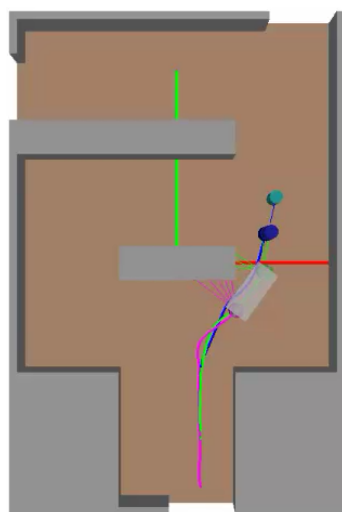
No cenário 6 (Figura 6.12a), existe a particularidade da equipa de robôs ter de seguir um alvo móvel (cor azul), que se desloca em direção a um alvo que está fixo (cor ciano). Neste cenário, a equipa de robôs parte segundo uma orientação de 90 graus, ao encontro do alvo móvel que parte da posição de coordenadas $(x, y, z) = (0.75, -3.25, 1)$, em direção ao alvo com coordenadas $(x, y, z) = (2.95, 2.0, 0)$.

Na Figura 6.12 observa-se o movimento realizado pela equipa de robôs desde a posição inicial até ao momento que o alvo móvel atinge o alvo que está fixo. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.12a), quando contorna o obstáculo (Figura 6.12b), presente na sala, e ainda quando o alvo móvel atinge o alvo fixo (Figura 6.12c).



(a) Equipa inicia o movimento.

(b) Equipa de robôs contorna o obstáculo.



(c) Alvo móvel atinge o alvo fixo.

Figura 6.12: Movimento da equipa até o alvo móvel atingir o alvo fixo.

Os resultados obtidos (Figura 6.13) demonstram que o algoritmo foi eficiente, ou seja, nunca foi comprometida a tarefa de transporte pois houve um comportamento correto por parte do robô *helper*.

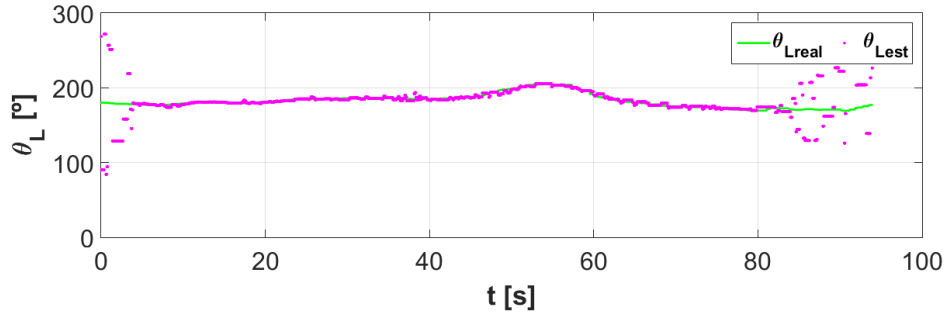


Figura 6.13: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 6.

6.1.1.7 Cenário 7

No cenário 7 (Figura 6.14a), para além da particularidade da equipa de robôs ter de seguir um alvo móvel (cor azul), que se desloca em direção a um alvo que está fixo (cor ciano), como no cenário anterior, o trajeto seguido pela equipa será diferente ao do alvo móvel. Esta diferença de trajetos deve-se ao facto, da passagem que se encontra no sentido do alvo fixo ser estreita, para a equipa de robôs atravessar, sem que comprometa a tarefa de transporte.

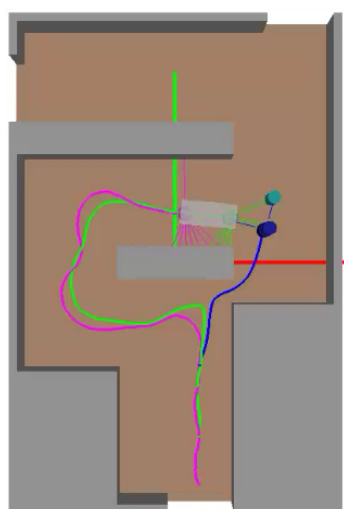
Neste cenário, a equipa de robôs parte segundo uma orientação de 90 graus, ao encontro do alvo móvel que parte da posição de coordenadas $(x, y, z) = (0.75, -3.25, 1)$, em direção ao alvo com coordenadas $(x, y, z) = (2.95, 2.0, 0)$.

Na Figura 6.14 observa-se o movimento realizado pela equipa de robôs desde da posição inicial até ao momento que a equipa atinge o alvo móvel. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.14a), quando contorna o obstáculo pelo lado contrário ao do alvo móvel (Figura 6.14b), presente na sala, e ainda quando a equipa atinge o alvo móvel (Figura 6.14c).



(a) Equipa inicia o movimento ($t = 0$ s).

(b) Equipa de robôs contorna o obstáculo pelo lado contrário ao do alvo móvel.



(c) Equipa de robôs atinge o alvo móvel.

Figura 6.14: Movimento da equipa até atingir o alvo móvel.

Os resultados obtidos (Figura 6.15) demonstram que o algoritmo foi eficiente, permitindo ao robô *helper* ter um comportamento correto na tarefa de transporte.

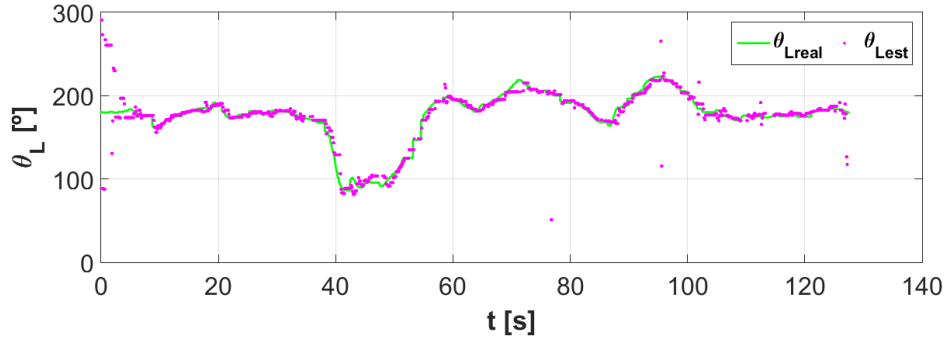
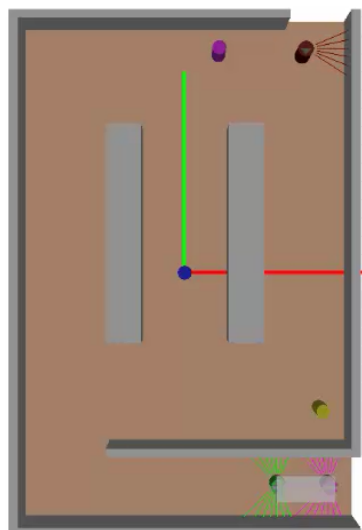


Figura 6.15: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 7.

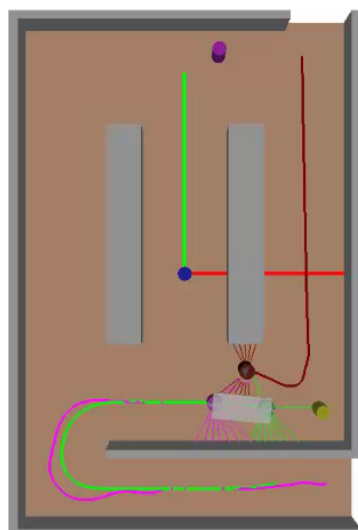
6.1.1.8 Cenário 8

No cenário 8 (Figura 6.16a), para além da particularidade de não existir apenas um alvo mas três alvos que a equipa tem de atingir, foi inserido um robô móvel que se desloca ao encontro de dois alvos (cor amarela e azul) presentes no cenário. A equipa de robôs após localizar o robô móvel desvia-se dele continuando e não comprometendo a tarefa de transporte que lhe foi confiada. Neste cenário, a equipa de robôs parte segundo uma orientação de 180 graus, ao encontro de um primeiro alvo (cor amarela) que se encontra na posição de coordenadas $(x, y, z) = (4, -4, 0)$, de um segundo alvo (cor rosa) que se encontra na posição de coordenadas $(x, y, z) = (1, 6.5, 0)$ e de um terceiro alvo (cor azul) na posição de coordenadas $(x, y, z) = (0, 0, 0)$.

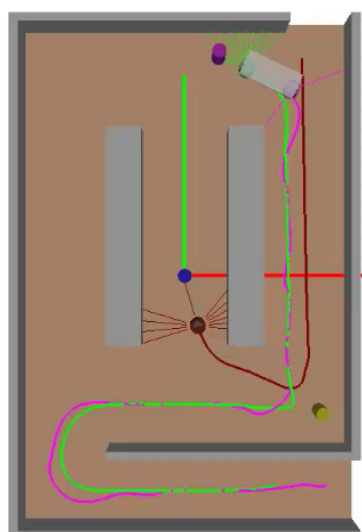
Na Figura 6.16 observa-se o movimento realizado pela equipa de robôs desde a posição inicial até ao momento em que atinge o último alvo. Nos instantes de tempo representados é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.16a), quando atinge o primeiro alvo e ao mesmo tempo se depara com a presença do robô móvel (Figura 6.16b) e ainda quando o atinge o segundo alvo (Figura 6.16c) e o último alvo presente no cenário (Figura 6.16d).



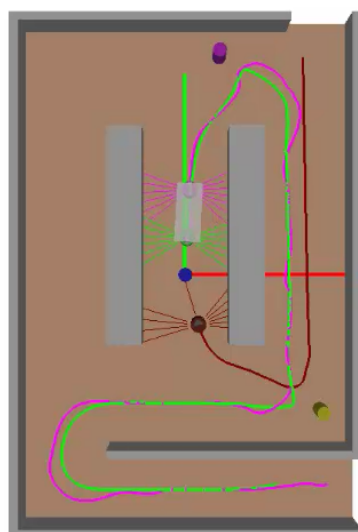
(a) Equipa inicia o movimento ($t = 0$ s).



(b) Equipa de robôs atinge o primeiro alvo e deteta a presença do robô móvel.



(c) Equipa de robôs atinge o segundo alvo.



(d) Equipa de robôs atinge o terceiro alvo.

Figura 6.16: Movimento da equipa até atingir o último alvo.

Os resultados obtidos (Figura 6.17) demonstram que o algoritmo teve um bom desempenho, refletindo um comportamento correto por parte do robô *helper* na tarefa de transporte.

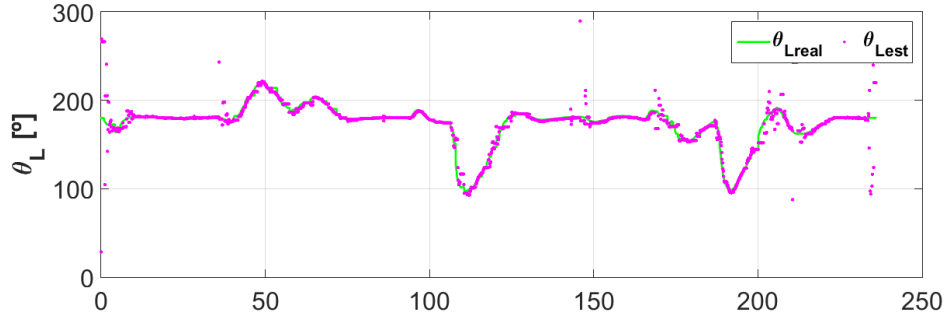


Figura 6.17: Ângulo que a carga faz com a direção atual do robô *leader* para o cenário 8.

6.1.2 Testes realizados no ambiente de simulação CoopDynSim utilizando controlo remoto

Os testes realizados no ambiente de simulação utilizando controlo remoto, para controlar o movimento do robô *leader* (velocidade linear e angular), permitiram avaliar a performance do algoritmo em situações que não são observáveis nos cenários. Nestas simulações o ambiente utilizado é simples correspondendo a uma sala retangular que não contem obstáculos nem alvos.

6.1.2.1 Situação 1

Na primeira situação, a equipa de robôs, realiza um movimento rotacional (Figura 6.18), perfazendo uma circunferência, com uma velocidade linear ($v_L = 0.20$ m/s) e angular ($w_L = 0.10$ rad/s) constante ao longo do movimento. Os resultados obtidos (Figura 6.19) demonstram que para esta situação o algoritmo teve um bom desempenho refletindo um comportamento correcto por parte do robô *helper* na tarefa de transporte.

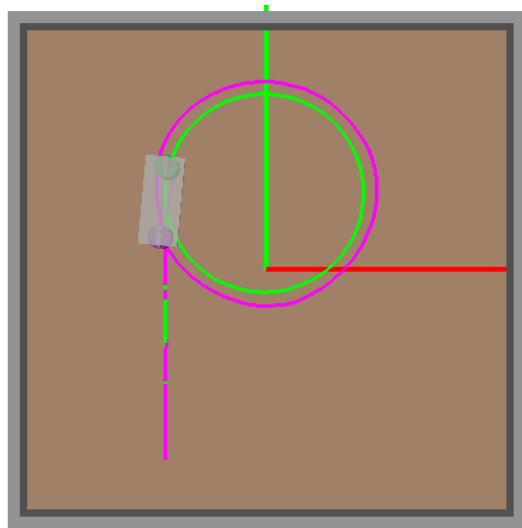


Figura 6.18: Movimento realizado pela equipa na situação 1.

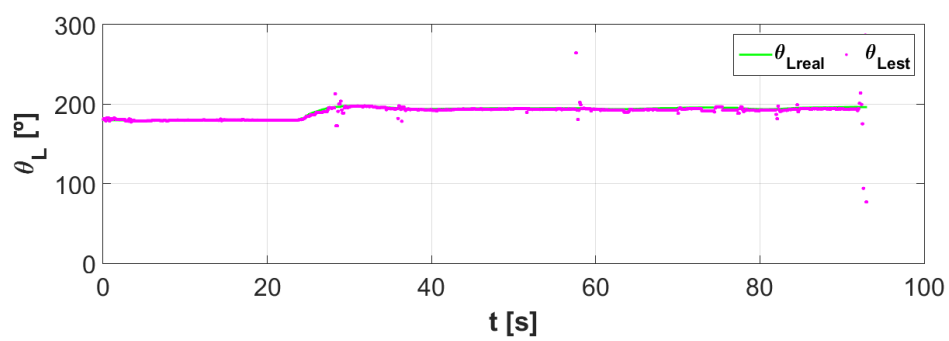


Figura 6.19: Ângulo que a carga faz com a direção atual do robô leader para a situação 1.

6.1.2.2 Situação 2

Na segunda situação, a equipa de robôs, realiza um movimento (Figura 6.20) com uma velocidade linear e angular que varia ao longo do trajeto.

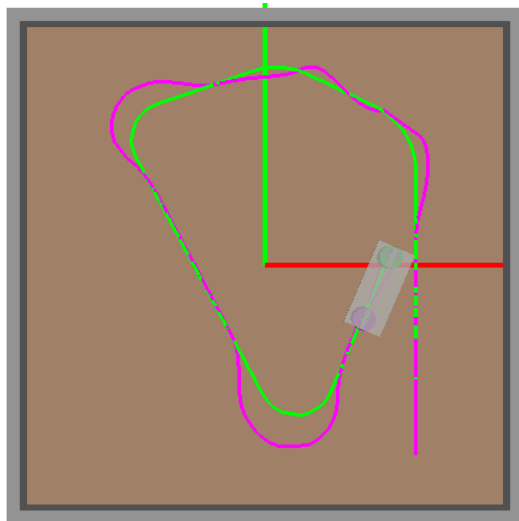


Figura 6.20: Movimento realizado pela equipa na situação 2.

Os resultados obtidos (Figura 6.21) demonstram que o algoritmo foi eficiente, ou seja, nunca foi comprometida a tarefa de transporte pois houve um comportamento correto por parte do robô *helper*.

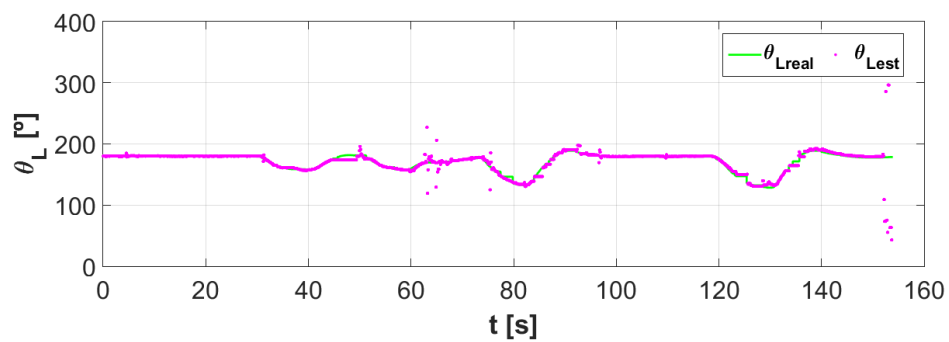


Figura 6.21: Ângulo que a carga faz com a direção atual do robô leader para a situação 2.

6.1.2.3 Situação 3

Na terceira situação, a equipa de robôs, realiza um movimento (Figura 6.22) com uma velocidade linear e angular que varia ao longo do trajeto. Nesta situação, contrariamente às situações anteriores, o robô leader teve uma variação tanto positiva como negativa na sua velocidade angular ao longo do movimento.

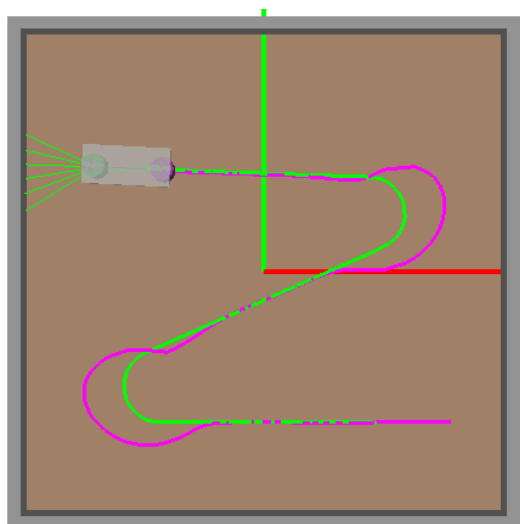


Figura 6.22: Movimento realizado pela equipa na situação 3.

Os resultados obtidos (Figura 6.23) demonstram que para esta situação o algoritmo teve um bom desempenho, refletindo um comportamento correto por parte do robô *helper* na tarefa de transporte.

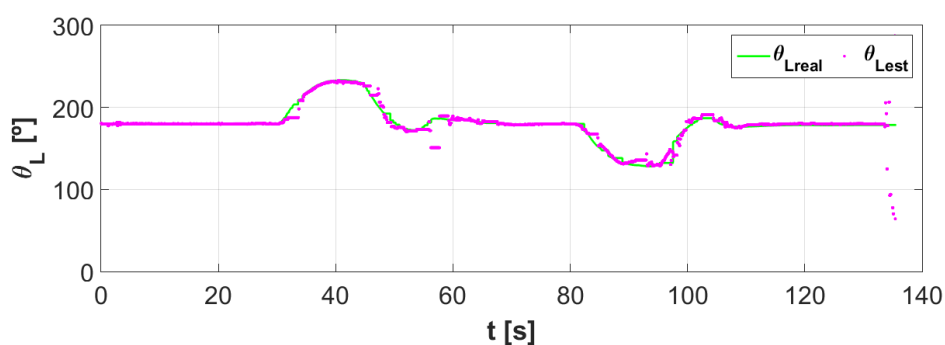


Figura 6.23: Ângulo que a carga faz com a direção atual do robô leader para a situação 3.

6.1.2.4 Situação 4

Na quarta situação pretendeu-se simular um comportamento por parte do robô *leader* que se assemelhasse a um realizado por um operador humano no comando da equipa. Nesse sentido, a trajetória (Figura 6.24) realizada pela equipa incluiu momentos de paragem, como se tratasse de tomadas de decisão por parte do operador humano na escolha do melhor caminho a seguir ou simplesmente pelo aparecimento de outros agentes no espaço.

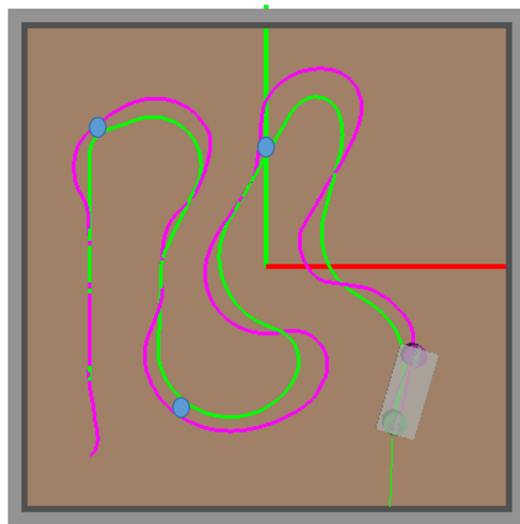


Figura 6.24: Movimento realizado pela equipa na situação 4.

Os resultados obtidos (Figura 6.25) demonstram que para esta situação o algoritmo teve um bom desempenho nos instantes de tempo que a equipa se encontrava em movimento. No entanto, nos instantes de tempo que a equipa realizou paragens (demarcadas na trajetória realizada pela equipa e limitados por linhas a tracejadas no gráfico) o algoritmo não foi capaz de fornecer uma medida do ângulo que a carga faz com a direção de navegação do robô *leader* correta, mas como foi abordado na seção 6.1.1.5 o algoritmo recupera com facilidade dessas incertezas. Porém, a tarefa de transporte foi bem sucedida.

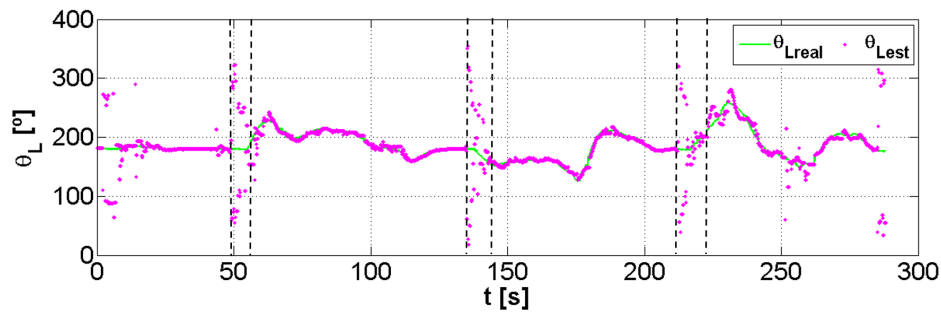


Figura 6.25: Ângulo que a carga faz com a direção atual do robô leader para a situação 4.

6.1.3 Testes realizados no ambiente real

Os testes realizados com os robôs reais foram efetuados nas instalações da Universidade do Minho. Na experiência realizada foi analisado o desempenho do módulo implementado na interação humano - robô.

Nesta experiência as dimensões da carga a transportar pela equipa humano - robô foram: 1.5 m de comprimento e 0.4 m de largura.

6.1.3.1 Interação humano - robô

Na Figura 6.26 é apresentado o cenário com as respetivas medidas e a trajetada o caminho desejado para o humano realizar. Neste cenário a equipa parte com uma orientação correspondente à da seta vermelha representada na Figura 6.26, que corresponde, de acordo com o referencial estipulado, a uma orientação de 90 graus.

Na Figura 6.27 está representado alguns dos instantes de tempo, onde é possível visualizar o momento em que a equipa inicia o movimento (Figura 6.27a), quando a equipa contorna um dos obstáculos (Figura 6.27b), presente no cenário, e ainda quando chega à posição destino (Figura 6.27c).

Nesta experiência, onde o robô *leader* foi substituído pelo humano, não se tem informação real da medida do ângulo que a carga faz com a direção do humano. Nesse sentido, na validação do desempenho do algoritmo implementado, no cálculo do ângulo que a carga faz com a direção atual do *leader*, é analisado os resultados obtidos para a trajetória do *helper* e a trajetória estimada para o operador humano por parte do robô *helper* (Figura 6.28).

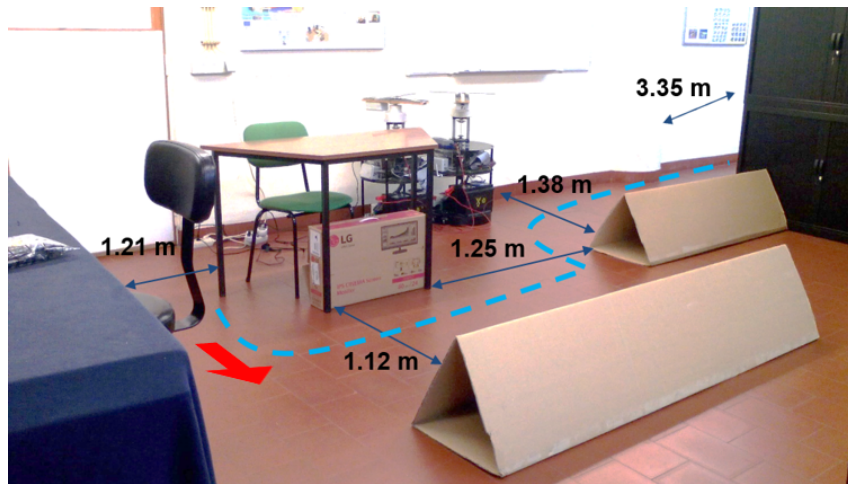


Figura 6.26: Planta do cenário.



(a) Equipa inicia o movimento ($t = 0$ s).



(b) Equipa de robôs contorna um dos obstáculos ($t = 19$ s).



(c) Equipa de robôs chega à posição destino ($t = 43$ s).

Figura 6.27: Movimento da equipa até chegar à posição de destino.

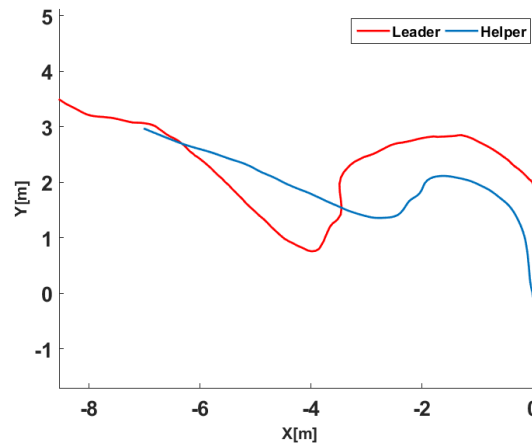


Figura 6.28: Trajetória realizada pela equipa humano - robô.

Observando os resultados obtidos para ambas as trajetórias verifica-se que a trajetória estimada para o operador humano aproxima-se à que foi efetivamente realizada (ver Figura 6.26).

Para além da trajetória também são analisados os resultados obtidos para a estimativa da direção de navegação do humano e para o ângulo que a carga faz com a direção de navegação do humano (Figura 6.29). Os resultados obtidos para a direção de navegação do humano refletem um comportamento por parte do humano coerente com o que foi efetivamente realizado. Isto é, inicialmente a estimativa da direção de navegação é um valor perto de 90 graus (excluindo os primeiros 3 s em que o algoritmo ainda não estabilizou), como foi definido inicialmente (Figura 6.27a), e vai variando até atingir um ponto máximo perto dos 270 graus, que se verifica quando a equipa contorna o obstáculo (Figura 6.27b), e a partir daí vai diminuindo até estabilizar num valor perto dos 180 graus que se verifica na fase final do percurso (Figura 6.27c). No caso do ângulo que a carga faz com a direção de navegação do humano os resultados obtidos aproximam-se do valor real na medida em que o valor é próximo dos 180 graus em situações que a carga está alinhada com a direção de navegação do humano, como é caso na fase inicial e final do percurso.

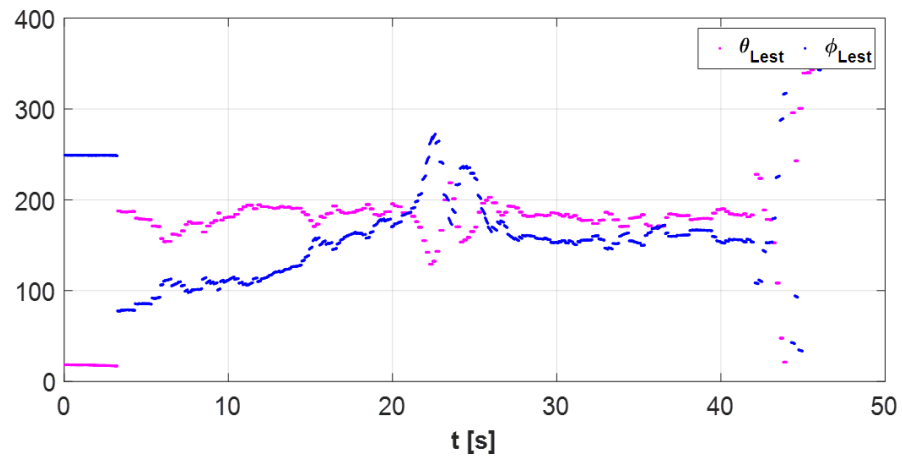


Figura 6.29: Direção de navegação do humano (ϕ_{Lest}) e o ângulo que a carga faz com a direção de navegação do humano (θ_{Lest}).

Capítulo 7

Conclusão

Nesta dissertação pretendia-se que uma equipa constituída por dois robôs móveis autónomos, um *leader* outro *helper*, transportassem cooperativamente uma carga de grandes dimensões de uma posição inicial até uma posição destino. Em trabalhos realizados, anteriormente, no Departamento de Eletrónica Industrial da Universidade do Minho foram desenvolvidas arquiteturas de controlo que permitem a uma equipa de dois ou mais robôs transportarem cooperativamente uma carga de grandes dimensões de uma posição inicial até a uma posição destino, evitando obstáculos ao mesmo tempo. Nessas arquiteturas, formalizadas recorrendo à Teoria de Sistemas Dinâmicos (na forma de equações diferenciais não lineares), o robô *helper* utiliza informação do robô *leader* para determinar a sua direção de navegação, no sentido de orientar a carga de modo que fique alinhada com a direção de navegação do robô *leader* e ao mesmo tempo evitando obstáculos. Nos trabalhos anteriores existia a necessidade de comunicação explícita para transmitir essa informação entre o robô *leader* e o robô *helper*.

Nos trabalhos realizados para esta dissertação pretendeu-se eliminar a comunicação entre o robô *leader* e o robô *helper*. Sendo assim, o objetivo principal desta dissertação foi dotar o robô *helper* de capacidades que lhe permitem estimar a informação sensorial necessária do robô *leader* para que desenvolva autonomamente a tarefa que lhe é proposta, permitindo assim eliminar a comunicação entre ele e o robô *leader*. Esta nova contribuição permite que o robô *leader* possa ser usado em ambientes em que a comunicação não é permitida, adiciona robustez em ambientes em que a comunicação é

ruidosa ou, ainda, permite que o robô *leader* possa ser substituído por um humano, de forma transparente do ponto de vista do robô *helper*, uma vez que este tem capacidades para agir de forma independente.

Inicialmente, começou-se pelo desenvolvimento do módulo que permite ao robô *helper* obter uma estimativa da informação sensorial necessária do *leader*, que neste caso é o ângulo que a carga faz com a direção de navegação do robô *leader*, e assim eliminar a comunicação que existe entre o *helper* e o *leader*. Para o desenvolvimento dessas capacidades recorreu-se ao cálculo trigonométrico e vetorial e ao filtro de Kalman Estendido.

Os resultados obtidos no ambiente de simulação demonstram que o algoritmo implementado permite eliminar a comunicação entre a equipa de robôs e conceder ao robô *helper* a capacidade de estimar a informação sensorial do *leader*. Nesse sentido, o robô *helper* teve um comportamento correto na realização da tarefa de transporte cumprindo a função de orientar a carga para que esta ficasse alinhada com a direção de navegação do robô *leader*. No ambiente real foi ainda possível validar o módulo implementado quando o *leader* é substituído pelo humano, verificando-se um comportamento correto por parte do robô *helper* na realização da tarefa de transporte. Os objetivos que se pretendia foram atingidos pois o robô *helper* respondeu de acordo com o esperado. Contudo, o módulo desenvolvido não foi capaz de dar uma medida do ângulo correta em situações em que a equipa possa parar ao longo do trajeto. Porém, essas limitações não prejudicam a realização da tarefa de transporte, como foi analisado nos testes efetuados. Nos resultados obtidos por vezes podem surgir erros de estimativa, mas como são filtrados pelo sistema dinâmico não afeta o comportamento do robô *helper* na tarefa de transporte.

Como trabalho futuro a desenvolver, neste âmbito, seria de interesse dar continuidade aos testes de interação com humanos para que se ajustem os movimentos, do robô, ao humano, e por isso haja melhor resposta daquele na tarefa de transporte. Nesse sentido, uma das hipóteses seria a implementação de um sistema de visão no robô *helper*, para aumentar a precisão da estimativa da direção seguida pelo operador humano quando este está no comando da equipa.

Bibliografia

- [1] Olli Alkkio, Ville Kyrki, Heikki Kälviäinen, Yong Liu, and Heikki Handroos. Complementing visual tracking of moving targets by fusion of tactile sensing. *Robotics and Autonomous Systems*, 57(11):1129–1139, 2009.
- [2] Yuichi Asahiro, Eric Chung-hui Chang, Ichiro Suzuki, Masafumi Yamashita, Communication Engineering, and Computer Science. A Distributed Ladder Transportation Algorithm for Two Robots in a Corridor. (2), 2001.
- [3] Christian Bettstetter. Collaborative mapping with mobile robot teams. <http://bettstetter.com/collaborative-mapping-with-mobile-robot-teams/>. Accessed: 2014-12-19.
- [4] Estela Bicho. *Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Shaker, 2000.
- [5] Robert Grover Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises and Solutions*. Wiley, 3 edition, 1996. ISBN 9780471128397.
- [6] Wolfram Burgard, Mark Moors, Cyrill Stachniss, Student Member, and Frank E Schneider. Coordinated Multi-Robot Exploration. 21(3):376–386, 2005.
- [7] Tran Huu Cong, Young Joong Kim, and Myo-Taeg Lim. Hybrid extended kalman filter-based localization with a highly accurate odometry model of a mobile robot. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 738–743. IEEE, 2008.

- [8] Husqvarna. <http://www.husqvarna.com/pt/products/robotic-mowers/automower-330x/>. Accessed: 2015-09-15.
- [9] Invacare. <http://www.invacare.pt/pt/content/invacare-stream-stream-1081pt>. Accessed: 2015-09-15.
- [10] iRobot. <http://www.irobot.com.br/Robos-domesticos/roomba-600>. Accessed: 2015-09-15.
- [11] Computation Laboratory for Advanced Sensing and Control. <http://ascc.okstate.edu/content/resources>. Accessed: 2014-12-19.
- [12] Martin Lawitzky, Jose Ramon Medina, Dongheui Lee, and Sandra Hirche. Feedback motion planning and learning from demonstration in physical robotic assistance: differences and synergies. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3646–3652, October 2012.
- [13] Toni Machado, Miguel Sousa, Sérgio Monteiro, and Estela Bicho. CoopDynSim : a 3D robotics simulator. *12th International Conference on Autonomous Robot Systems and Competitions*, pages 45–51, 2012.
- [14] Toni Machado, Tiago Malheiro, Sergio Monteiro, Estela Bicho, and Wolfram Erlhagen. Transportation of long objects in unknown cluttered environments by a team of robots: A dynamical systems approach. *2013 IEEE International Symposium on Industrial Electronics*, pages 1–6, May 2013.
- [15] Jose Ramon Medina, Martin Lawitzky, Alexander Mortl, Dongheui Lee, and Sandra Hirche. An experience-driven robotic assistant acquiring human knowledge to improve haptic cooperation. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2416–2422, September 2011.
- [16] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. Yarp: yet another robot platform. *International Journal on Advanced Robotics Systems*, 3(1):43–48, 2006.

- [17] Hyeun Jeong Min, Andrew Drenner, and Nikolaos Papanikolopoulos. Vision-based leader-follower formations with limited information. *2009 IEEE International Conference on Robotics and Automation*, pages 351–356, May 2009.
- [18] Alexander Mortl, Martin Lawitzky, Ayse Kucukyilmaz, Metin Sezgin, Cagatay Basdogan, and Sandra Hirche. The role of roles: Physical cooperation between humans and robots. *The International Journal of Robotics Research*, 31(13):1656–1674, August 2012.
- [19] Guilherme A.S. Pereira, Bruno S. Pimentel, Luiz Chaimowicz, and Mario F.M. Campos. Coordination of multiple mobile robots in an object carrying task using implicit communication. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, 1:281–286, 2002.
- [20] Sameera Poduri and Gaurav S. Sukhatme. Constrained coverage for mobile sensor networks. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 1, 2004.
- [21] Robotnik. <http://www.robotnik.eu/mobile-robots/autonomous-agvs/>. Accessed: 2015-09-15.
- [22] DongBin Shen, ZhenDong Sun, and WeiJie Sun. Leader-follower formation control without leader’s velocity information. *Science China Information Sciences*, 57(9): 1–12, July 2014.
- [23] Rui Soares, Estela Bicho, Toni Machado, and Wolfram Erlhagen. Object transportation by multiple mobile robots controlled by attractor dynamics: Theory and implementation. *IEEE International Conference on Intelligent Robots and Systems*, pages 937–944, 2007.
- [24] J Zhang, Y Wu, W Liu, and X Chen. Novel approach to position and orientation estimation in vision-based uav navigation. *Aerospace and Electronic Systems, IEEE Transactions on*, 46(2):687–700, 2010.